Universidade do Minho

Gonçalves, David

Processo de replicação de cenários e automatização de testes em redes IMS

http://hdl.handle.net/1822/27951

Metadados

Data de Publicação 2013-12-11

Resumo IP networks are presented as the main communication infrastructures used for an extensive range of services, which include voice services previously provided by legacy networks. The shift from circuit-switched networks to packet-switched networks was driven by the new paradigms and architectures of telecommunications networks, such as the IMS networks. With IMS it becomes possible to provide a wide range of services quickly, and aligned with the users expectations. The rapid availability of...

As redes IP apresentam-se como as principiais infraestruturas de comunicação usadas para um conjunto extenso de serviços, nos quais se incluem os serviços de voz anteriormente disponibilizados pelas redes legadas. A passagem dos serviços de voz das redes comutadas de circuitos para as redes comutadas de pacotes foi impulsionada por novos paradigmas e arquiteturas das redes de telecomunicações, sendo um desses casos as redes IMS. Através das redes IMS tornase possível a disponibilização de u...

Tipo masterThesis

Esta página foi gerada automaticamente em 2015-05-15T18:34:32Z com informação proveniente do RepositóriUM

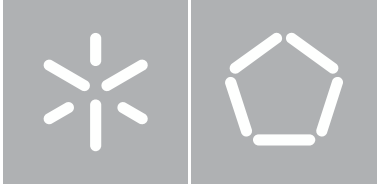
http://repositorium.sdum.uminho.pt

Universidade do Minho
Escola de Engenharia

David Almeida Gonçalves

Processo de Replicação de Cenários e Automatização de Testes em Redes IMS

Outubro de 2013



Universidade do Minho
Escola de Engenharia
Departamento de Informática

David Almeida Gonçalves

Processo de Replicação de Cenários e Automatização de Testes em Redes IMS

Dissertação de Mestrado
Mestrado em Redes e Serviços de Comunicações

Trabalho realizado sob orientação de

Professor Dr. António Duarte Costa
Professor Dr. Pedro Nuno Miranda de Sousa
Mestre António Manuel Amaral

Outubro de 2013

Agradecimentos

Ao Professor António Duarte Costa, pela orientação, disponibilidade e preocupação. Sem esse apoio ainda agora não teria a dissertação como pretendia.

Ao Professor Pedro Nuno Sousa, por me ajudar mesmo antes da primeira palavra estar escrita até ao último ponto ser colocado.

Ao António Amaral, por me orientar e, acima e tudo, por me levar a ser melhor e a querer fazer sempre melhor. Com todos os desafios propostos aprendi e cresci a vários níveis.

A todos os colegas da PT Inovação que me ajudaram, em especial ao José Carlos Amorim que ao longo de todo o processo ensinou-me mais do que eu era capaz de aprender, sendo um exemplo de paciência e dedicação, capaz de me auxiliar sempre com um sorriso na cara.

Aos meus amigos que me acompanharam durante toda a dissertação, levando-me a acreditar que as dificuldades eram passageiras e a solução apareceria.

Finalmente à minha família pelo apoio dado e por me ajudarem a chegar até aqui, auxiliando-me em tudo que precisei ao longo do tempo.

A todos o meu sincero obrigado.

Abstract

IP networks are presented as the main communication infrastructures used for an extensive range of services, which include voice services previously provided by legacy networks. The shift from circuit-switched networks to packet-switched networks was driven by the new paradigms and architectures of telecommunications networks, such as the IMS networks. With IMS it becomes possible to provide a wide range of services quickly, and aligned with the users expectations.

The rapid availability of services is one of the main principles of the IMS network, so it is essential to develop methods and mechanisms that convert the manual validation processes into automatic tasks of expedit execution. With this objective outlined, it was necessary to study the manual validation processes and to analyze protocols and adjacent equipment with the objective to create procedures or methods that automate much of the validation process.

This dissertation presents a possible solution for automated validation of SIP solutions in IMS networks, highlighting the gains that the solution presents compared to the manual validation processes. The proposal is supported by the study of the IMS network and by the analysis and execution of manual validation processes in several scenarios. On these processes, the main components of the IMS architectural and functional levels are analyzed. As a result, with the proposed solution it is possible to drastically reduce validation times when compared with the corresponding manual approaches. Moreover, the devised automated solution allows to enhance testing reliability and coverage.

Resumo

As redes IP apresentam-se como as principiais infraestruturas de comunicação usadas para um conjunto extenso de serviços, nos quais se incluem os serviços de voz anteriormente disponibilizados pelas redes legadas. A passagem dos serviços de voz das redes comutadas de circuitos para as redes comutadas de pacotes foi impulsionada por novos paradigmas e arquiteturas das redes de telecomunicações, sendo um desses casos as redes IMS. Através das redes IMS torna-se possível a disponibilização de um conjunto alargado de serviços de forma célere, indo de encontro ao desejado pelos utilizadores da rede.

A rápida disponibilização de serviços é um dos princípios base das redes IMS, pelo que é fundamental desenvolver métodos e mecanismos capazes de tornar os processos de validação em tarefas automáticas e de rápida execução. Com o objetivo de agilizar os processos de validação, iniciou-se o estudo dos processos manuais, analisando protocolos e equipamentos adjacentes, tendo como meta criar procedimentos ou métodos capazes de automatizar grande parte dos processos de validação.

Nesta dissertação apresenta-se uma proposta de validação automática de soluções SIP nas redes IMS, destacando as mais-valias que a solução apresenta em relação aos processos de validação manual. A proposta apresentada é sustentada pelo estudo realizado sobre as redes IMS e pela análise e execução dos processos de validação manual de diversas soluções. Nestes processos são analisados os principiais componentes da arquitetura IMS a nível funcional e arquitetural. A proposta desenvolvida apresenta tempos de validação muito inferiores aos decorrentes do processo manual, assegurando também um maior nível de cobertura e confiabilidade nos testes de validação de equipamentos IMS.

Índice

Lista de Figuras **xiii**

Lista de Tabelas **xv**

Lista de Acrónimos **xvii**

1 Introdução **1**

1.1 Motivação 2

1.2 Objetivos 4

1.3 Resumo dos principais contributos 4

1.4 Organização da Dissertação 5

2 Estado da Arte **7**

2.1 Da 1ª à Próxima Geração 7

2.2 Arquitetura das Redes de Próxima Geração 12

2.2.1 ITU-T NGN 12

2.2.2 ETSI TISPAN 14

2.3 IP Multimedia Subsystem 15

2.3.1 Arquitetura IMS 17

2.3.2 Protocolos IMS 23

2.4 Resumo 30

3 Automatização de Testes **31**

3.1 Taxonomia de Testes 31

3.1.1 Testes de Compatibilidade 31

3.1.2 Testes de Sanidade 32

3.1.3 Testes de Regressão 32

3.1.4 Testes de Aceitação 32

3.1.5 Testes de Desempenho 33

3.2 Automatização de Testes SIP 33

3.2.1 Criação de Cenários 35

3.2.2 Criação de Testes 36

3.2.3 Execução de Testes 40

3.3 Equipamentos sob Processo de Automatização 41

3.3.1 Session Border Controller 41

3.3.2 F5 BIG-IP 44

3.4 Resumo 45

4 Replicação de Cenários e Automatização de Testes: Solução Proposta **47**

4.1 Arquitetura global do Sistema de testes 47

4.2 Emulação de tráfego SIP 48

4.2.1 SIPp 49

4.3 Preparação do cenário de teste 52

4.3.1 SIP-Scenario 54

4.3.2 *Scripts* de Arranque 57

4.4 Processo de criação dos testes 58

4.4.1 sniff2Sipp 60

4.5 Metodologia de execução dos testes 62

4.5.1 SIPp-ATester 63

4.6 Resumo 68

5 Teste e validação da solução proposta **69**

5.1 Demonstração de funcionalidades 69

5.1.1 Criação dos testes 69

5.1.2 Criação do cenário 72

5.1.3 Execução dos testes 73

5.2 Resultados obtidos e aplicabilidade 75

5.3 Resumo 77

6 Conclusão **79**

6.1 Trabalho Futuro 80

A Revista Saber & Fazer - Automatização de testes em Redes IMS **81**

B Mapeamento de requisitos em testes: Caso prático **93**

C Análise comparativa entre pcap2sipp e sniff2sipp **95**

D Template Expressões regulares **97**

D.1 Validação da existência do cabeçalho 97

D.2 Validação de valores estáticos 98

D.3 Validação de valores dinâmicos 99

Bibliografia **99**

Lista de Figuras

2.1 Modelo funcional definido pela ITU-T [1] 13

2.2 Arquitetura estratificada IMS [2] 15

2.3 Compatativo entre arquitetura de integração de serviços das redes legadas e das redes IMS [3] 17

2.4 Arquitetura da rede IMS 19

2.5 Conversa telefónica com fluxo de sinalização similar ao presente em SIP 25

2.6 Mapeamento de um conversa num fluxo SIP 25

2.7 Protocolo Diameter base e aplicações 30

3.1 Arquitetura típica de rede IMS presente na rede do cliente 35

3.2 Modelo seguido na rede de desenvolvimento para validação do equipamento . . . 37

3.3 Estabelecimento de sessão entre dois terminais na mesma *home network* 38

3.4 Fluxo inicial tido no estabelecimento de uma sessão SIP 38

3.5 Mensagem entre o I-CSCF2 e o S-CSCF2 (input do S-CSCF2) 39

3.6 Mensagem entre o S-CSCF2 e o P-CSCF2 (output do S-CSCF2) 40

3.7 Validação do *user part* do cabeçalho From 40

3.8 Ramos de configuração do SBC 42

3.9 Configurações essenciais no SBC 43

3.10 Diagrama de arquitetura *peer-to-peer* 44

3.11 Diagrama de arquitetura F5 BIG-IP 45

4.1 Diagrama de componentes da solução 48
4.2 Fluxos no cenário de chamada por defeito do SIPp 49
4.3 Comparação do conteúdo da *tag send* com o presente na *tag recv* 50
4.4 Método INVITE presente no *script* SIPp 50
4.5 Exemplo de configuração de expressão regular 51
4.6 Comandos de arranque dos scripts SIPp 51
4.7 Shell de execução do UAC SIPp 52
4.8 Procedimento para a criação de cenários para teste 53
4.9 Arquitetura de rede de equipamento SBC 54
4.10 Diagrama de sequência da aplicação SIP-Scenario 55
4.11 Processo de filtragem de documentos XML 56
4.12 Comandos executados para validação das configurações do BIG-IP 56
4.13 Script exemplo de cenário 57
4.14 Comandos associados à execução do *script* exemplo de cenário 57
4.15 Procedimento para a criação de testes 58
4.16 Conteúdo de ficheiro pcap capturado na rede do cliente 59
4.17 Mensagem SIPp criada através do conteúdo do ficheiro pcap 59
4.18 Código de tratamento de VLANs 61
4.19 Comportamento por defeito da aplicação sniff2sipp 61
4.20 Substituição dos valores definidos no -v por valores field*n* 62
4.21 Procedimento para a execução de testes 63
4.22 Diagrama de blocos da aplicação SIPp-ATester 64
4.23 Diagrama de sequência da aplicação SIPp-ATester 65
4.24 Script tipo usado para executar os scripts SIPp 67
4.25 Comando usado para a execução do Shell script 68

5.1 Fluxo de mensagens trocadas entre o UAC e UAS sobre o equipamento a validar 70

5.2 Comando perl para execução da aplicação sniff2sipp 70
5.3 Output da aplicação sniff2sipp 70
5.4 Conteúdo da mensagem INVITE presente na captura de rede do cliente 71
5.5 *Script* SIPp gerado através da captura da rede do cliente 71
5.6 Relatório do estado das configurações obtido pelo SIP-Scenario 72
5.7 Script de arranque e comandos executados no equipamento 73
5.8 Output intermédio da aplicação SIPp-ATester 73
5.9 Relatório final da aplicação SIPp-ATester 74
5.10 Expressão regular aplicada ao teste SIPp falhado 74
5.11 Evidência do teste falhado 75
5.12 Gráfico de tempos despendidos em validações de firmware 76

B.1 Configuração modelo de um teste 94

C.1 Inicialização de *scripts* pcap2sipp 96
C.2 Inicialização de *scripts* sniff2sipp 96

D.1 Captura do valores associados ao P-Preferred-Identity 97
D.2 Comparação dos valores do P-Preferred-Identity 98
D.3 Validação do valor obtido da comparação dos valores do P-Preferred-Identity . . 98
D.4 Validação dos valores de emergência no RURI do INVITE 98
D.5 Obtenção do user-part e host-part do cabeçalho From 99
D.6 Comparação dos valores anteriormente obtidos com valores de ficheiro 99
D.7 Validação dos valores obtidos da comparação dos valores dinâmicos 99

Lista de Tabelas

2.1 Métodos associados com a RFC base do SIP 26

2.2 Status associados com a RFC base do SIP 26

2.3 Protocolo SDP: Informação da descrição da sessão [4] 28

2.4 Protocolo SDP: Informação da descrição temporal [4] 28

2.5 Protocolo SDP: Informação de media [4] 29

4.1 Informação do cenário a testar 66

4.2 Informação de autenticação de UAC e UAS 66

4.3 Parâmetros usados no SIPp 66

B.1 Testes associados com o requisito 93

C.1 Vantagens e desvantagens do pcap2sipp 95

C.2 Vantagens e desvantagens do sniff2sipp 96

Lista de Acrónimos

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ABNF	Augmented Backus-Naur Form
AMPS	Advanced Mobile Phone System
AVP	Attribute-Value Pair
AS	Application Server
BSS	Base Station System
BGCF	Breakout Gateway Control Function
BICC	Bearer-Independent Call Control
CAMEL	Customized Applications for Mobile network Enhanced Logic
CDMA	Code-Division Multiple Access
CS	Circuit-Switching
CSCF	Call Session Control Function
DNS	Domain Name System
DSCP	Diffserv Code Point
ETSI	European Telecommunications Standards Institute
FDMA	Frequency-Division Multiple Access
FTP	File Transfer Protocol
GPRS	General Packet Radio Service

GSM	Global System for Mobile Communications
HLR	Home Location Register
HSS	Home Subscriber Server
IETF	Internet Engineering Task Force
IFC	Initial Filter Criteria
IP	Internet Protocol
IPSec	IP Security Protocol
IP-CAN	IP Connectivity Access Network
IM	IP Multimedia
IS-95	Interim Standard 95
ISUP	ISDN User Part
LTE	Long-term Evolution
LTM	Local Traffic Manager
LRF	Location Retrieval Function
IMS	IP Multimedia Subsystem
IMS-ALG	IMS Application Layer Gateway
IM-SSF	IP Multimedia Service Switching Function
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
MG	Media Gateway
MRFC	Media Resource Function Controller
MRFP	Media Resource Function Processor
NGN	Next-Generation Network
NTP	Network Time Protocol
OSA-SCS	Open Service Access-Service Capability Servers
PCM	Pulse Code Modulation

PDP	Packet Data Protocol
PS	Packet-Switching
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request for Comments
RPG	Redes de Próxima Geração
RTP	Real-time Transport Protocol
RTCP	RTP Control Protocol
SBC	Session Border Controller
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SGW	Signaling Gateway
SSH	Secure Shell
SSL	Secure Sockets Layer
TACS	Total Access Communication System
TCP	Transmission Control Protocol
TDMA	Time-Division Multiple Access
THIG	Topology Hiding Inter-network Gateway
TISPAN	Telecoms & Internet converged Services & Protocols for Advanced Network
TrGW	Transition Gateway
UDP	User Datagram Protocol
UE	User Entity
URI	Uniform Resource Identifier
URI	Uniform resource identifier
VLAN	Virtual Local Area Network

Capítulo 1

Introdução

Ao longo das últimas décadas as redes de telecomunicações têm apresentado um crescimento exponencial, tanto devido aos melhoramentos tecnológicos como ao constante desejo por parte dos utilizadores finais de possuírem serviços melhorados e cada vez mais ubíquos.

Nas redes de telecomunicações evidenciam-se duas principais vertentes, as redes de telefonia e as redes de dados. Aquando da criação de ambas eram claras as diferenças existentes e os serviços prestados eram claramente delimitados com fronteiras de atuação bem definidas. Contudo, à medida que as redes de telecomunicações evoluíam, os serviços prestados por ambas as redes convergiam, sendo na atualidade difícil indicar que serviços se encontram só associados às redes de telefonia ou às redes de dados.

A convergência entre os serviços levanta questões sobre a necessidade de se continuar a manter duas redes distintas para a prestação de serviços similares. Estas questões apresentam-se pertinentes, verificando-se benefícios económicos e tecnológicos associados à aglutinação dos serviços sobre uma única infraestrutura. De forma geral, tornou-se evidente que a evolução das redes de telecomunicações passaria pelo desenvolvimento de uma rede convergente capaz de dar resposta às necessidades dos utilizadores. Os primeiros passos no desenvolvimento das redes convergentes foram iniciados pela International Telecommunication Union Telecommunication Standardization Sector (ITU-T). A entidade de normalização juntou esforços e iniciou o processo de conceção dos princípios associados às chamadas Redes de Próxima Geração (RPG), designadas em inglês por Next-Generation Network (NGN).

Em 2004 foram disponibilizados pela ITU-T os princípios gerais e o modelo de referência das redes NGN nas normalizações Y.2001 [5] e Y.2011 [6]. As duas normalizações apresentavam-se

como a base a seguir para a implementação de redes NGN. Contudo, não identificam protocolos nem equipamentos a utilizar, mostrando as redes NGN de forma mais abstrata, sendo dado especial enfâse aos princípios base. Estes princípios assentavam na capacidade de garantir os requisitos de Qualidade de Serviço (QoS) e mobilidade expectáveis pelos utilizadores, disponibilizando um conjunto de serviços independentes do terminal, garantindo sempre elevada largura de banda.

Seguindo os princípios definidos pela ITU-T, é lançado no início de 2006 sobre a alçada da European Telecommunications Standards Institute (ETSI) Telecoms Internet converged Services Protocols for Advanced Network (TISPAN) um modelo normalizado de rede NGN [7]. Neste modelo, a TISPAN apresenta as redes NGN como redes convergentes capazes de dar resposta aos múltiplos serviços disponibilizados sobre uma plataforma IP. Em termos estruturais, o modelo definia a rede como tendo uma arquitetura modular constituída por múltiplos subsistemas normalizados, sendo um dos principiais subsistemas o IP Multimedia Subsystem (IMS) [8] definido pelo 3rd Generation Partnership Project (3GPP).

O IMS apresenta-se como uma arquitetura funcional normalizada que tem por objetivo a disponibilização de serviços sobre uma plataforma única a terminais associados a redes distintas. Caracteriza-se como sendo uma arquitetura dividida em camadas horizontais, apresentando independência entre cada uma, separando toda a sua arquitetura na camada de transporte, controlo e aplicação.

Em termos aplicacionais, as redes NGN disponibilizam um conjunto de serviços independentes do terminal e da localização, permitindo assim, a criação de um sistema verdadeiramente ubíquo, indo de encontro ao desejado pelos utilizadores. Do ponto de vista dos operadores, as redes NGN apresentam-se como tendo a capacidade de libertação dos tradicionais serviços de voz, dando a possibilidade de disponibilização de um conjunto de serviços alargado com modelos de taxação adequados. Oferecendo vantagens para utilizadores e operadores as redes NGN afiguram-se como sendo o passo natural na evolução das comunicações.

1.1 Motivação

As NGN apresentam-se como o passo a dar pelos operadores de telecomunicações na evolução das redes. Contudo, a validação de uma arquitetura complexa possuidora de um conjunto de elementos diversificados, acarreta um leque de problemas de elevada complexidade.

Os problemas evidenciados na validação de redes NGN verificam-se também na sua maioria nas redes IMS. A complexidade das funcionalidades presentes, bem como o conjunto diversificado de protocolos, colocam muitos desafios ao processo de validação. Ao nível protocolar, nas redes IMS destacam-se três protocolos, o Session Initiation Protocol (SIP) [9], o Diameter [10] e o Real-time Transport Protocol (RTP) [11], sendo necessário para cada um dos protocolos realizar um conjunto de testes específicos sobre cada equipamento onde os protocolos operam. Do ponto de vista dos equipamentos existe a necessidade de realizar validações em cada cenário onde o equipamento se encontra presente. A necessidade de analisar o comportamento dos equipamentos nos diferentes cenários verifica-se pelo facto de o mesmo fluxo de mensagens poder gerar diferentes comportamentos consoante o cenário de execução. Este comportamento verifica-se porque, a grande parte dos equipamentos são capazes de realizar manipulações sobre o conteúdo trocado pelos diferentes protocolos, fazendo manipulações consoante a origem, destino, tipo de sessão, entre outros.

Juntamente com as questões relacionadas com a diversidade de testes verificam-se também problemas relacionados com a frequência com que estes são executados. De forma a ter a solução validada é necessário realizar verificações dos testes sempre que surgem novas versões de configurações, aquando de atualizações de *firmware* e quando se realizam modificações de *hardware* sendo necessário efetuar para cada uma destas situações um conjunto parcial ou total de testes da solução.

Em adição às problemáticas identificadas anteriormente, verificam-se também questões associadas com a própria execução dos testes. As configurações a validar, por norma, encontram-se implementadas em equipamentos de produção, tendo o equipamento grande iteração com os elementos circundantes, iterações essas que devem ser levadas em conta no momento de validação do equipamento.

Todas as condicionantes identificadas tornam o processo de validação árduo e dispendioso tanto a nível de recursos como de tempo, verificando-se a necessidade de automatizar o processo de validação. A automatização apresenta-se como um passo natural no desenvolvimento de soluções e execução de tarefas rotineiras. Contudo, todo o processo de automatização possui dificuldades intrínsecas, não sendo a automatização de testes IMS exceção. Com o esforço tido no processo de automatização, pretende-se desenvolver mecanismos que permitam replicar as configurações de equipamentos presentes nas redes dos clientes (redes de operadores que compram as soluções IMS), desenvolver mecanismos para a geração de testes de forma rápida e intuitiva e criar métodos de execução de testes para validação de baterias de testes de forma a

obter evidências dos comportamentos.

1.2 Objetivos

Os objetivos deste trabalho estão associados ao estudo e desenvolvimento de uma solução de automatização capaz de ser aplicada aos diferentes elementos da rede IMS, dando particular destaque aos equipamentos Session Border Controllers (SBC) da Oracle e aos *Load Balancers* BIG-IP da F5. Neste contexto será necessário ao longo deste trabalho atingir os seguintes objetivos parciais:

- Conhecimento do funcionamento das redes IMS e dos protocolos SIP e Diameter.
- Desenvolvimento de competências de configuração de equipamentos SBCs e Load Balancers BIG-IP.
- Exploração de ferramentas de emulação de tráfego de redes IMS para os diferentes protocolos.
- Formulação e desenho de uma solução de automatização de testes para as redes IMS.
- Desenvolvimento e validação da solução de automatização de testes.

1.3 Resumo dos principais contributos

O trabalho desenvolvido tem valor a nível empresarial, possuindo uma elevada componente prática voltada para a resolução de um problema real. Contudo, os contributos da solução apresentam-se também úteis para uma grande variedade de soluções e temas de estudo.

Ao nível de desenvolvimento, o trabalho realizado pode ser aplicado a um conjunto vasto de equipas multidisciplinares que tenham contacto direto com redes IMS, servindo o trabalho não só para a realização das fases de testes, mas também para a simulação de cenários anómalos, avaliação performativa de equipamentos e análise estatística de comportamentos.

Ao nível de investigação, a utilização da plataforma de testes é útil para o estudo comparativo de soluções IMS, podendo validar comportamentos e desempenhos de diferentes soluções de forma célere. De igual forma, a plataforma será igualmente capaz de auxiliar na simulação de cenários reais, facilitando a tarefa de emulação de tráfego o mais próximo possível da realidade.

No que diz respeito à disseminação de conhecimento, o trabalho apresenta duas publicações associadas, sendo uma delas um resumo alargado publicado na Conferência de Redes e Computadores (CRC 2013) [12] e um artigo publicado na revista interna da PT Inovação Fazer & Saber, que se encontra presente no apêndice A.

1.4 Organização da Dissertação

No capítulo presente, é identificada de forma breve a problemática associada à realização de testes sobre as redes IMS, apresentando a motivação e os objetivos subjacentes ao desenvolvimento do trabalho.

No Capítulo 2 é descrita a evolução das redes de telecomunicações ao longo das últimas décadas, dando especial enfâse às NGN e às redes IMS nesta evolução. Ao nível das NGN são explicadas as motivações que levaram ao seu surgimento e quais os problemas que estas endereçam através de diferentes normalizações. Relativamente às redes IMS, é explicado o seu funcionamento, identificando protocolos e componentes arquiteturais.

O Capítulo 3 analisa todas as questões a endereçar na proposta a construir, sendo identificadas as diferentes fases de testes existentes e os processos subjacentes à criação de uma solução de automatização. Para além da identificação das problemáticas, são identificados os equipamentos sobre os quais a proposta vai incidir na fase de validação e teste.

A proposta de automatização desenvolvida é apresentada no Capítulo 4, onde é identificada a arquitetura da solução, descritos os procedimentos de validação e explicadas as ferramentas e aplicações usadas para agilizar o processo de validação.

No Capítulo 5 é apresentada a execução da proposta e os seus benefícios. Inicialmente são explicadas as diferentes fases do processo de validação automática, demonstrando o processo de validação desde a criação de testes até à execução dos mesmos. No final do capítulo é feita a comparação da validação manual com a automática demonstrando as diferenças e os benefícios da proposta.

Finalmente, no Capítulo 6, conclui-se o trabalho com a avaliação e reflexão dos resultados obtidos, apontando o possível trabalho futuro a realizar.

Capítulo 2

Estado da Arte

Este capítulo aborda a temática da evolução das redes de telecomunicações e o papel do IMS nessa mesma evolução. Inicialmente apresenta-se a evolução das redes na secção 2.1. De seguida, na secção 2.2 apresentam-se as normalizações das redes NGN destacando o papel destas. No final, descreve-se o IMS explicando o seu funcionamento, arquitetura e protocolos.

2.1 Da 1ª à Próxima Geração

A comunicação sempre fez parte da vida do ser humano, tendo sofrido contínuas evoluções para acompanhar as mudanças vivenciadas pelo homem, refletidas também nas mudanças observadas nas redes de telecomunicações. Desde o aparecimento do telégrafo [13], as redes de telecomunicações têm evoluído a um ritmo exponencial, acompanhando a procura pela partilha e acesso à informação requerida pelos seus utilizadores.

Em termos léxicos a palavra telecomunicação surge do grego *tele* que significa distância e do termo latim *communicatio* que significa "conexão" [14], ou seja a capacidade de criar uma ligação com outra entidade que esteja numa localização distante. Esta conexão tem como propósito a transferência de informação, podendo ocorrer sobre diferentes tipos de meios (ligações eletromagnéticas, ligações rádio, ligações infravermelhos).

Como referido inicialmente, a evolução das redes de telecomunicações teve o seu início no desenvolvimento do telégrafo. De entre as diferentes versões, destaca-se o telégrafo de Morse que surgiu em 1837 [15]. Através do telégrafo tornou-se possível a comunicação a longas distâncias atingindo os objetivos primários das redes de telecomunicações. Contudo, sendo um sistema

baseado em padrões, não se apresentava como um sistema intuitivo e de fácil utilização.

A necessidade de criar um sistema intuitivo, capaz de permitir comunicações em simultâneo, deu mote ao surgimento do telefone. Em 1876, inventado por Alexandre Graham Bell's, surge o primeiro telefone [16]. O telefone de Bell operava fazendo a codificação no originador da voz em sinais elétricos, que eram transmitidos por ligações de cabo sobre distâncias consideráveis. No recetor era realizada a descodificação do sinal de forma a tornar a voz audível novamente. Em termos funcionais, o telefone de Bell funcionava tirando proveito princípios de magnetismo, utilizando uma bobine em conjunto com uma membrana para criar flutuações na corrente. Após a invenção do telefone, as primeiras palavras ditas foram para o assistente de Bell, Thomas Watson, indicando que Bell o queria ver *'Mr. Watson-come here-I want to see you.'* [17].

Com os avanços apresentados pelo telefone tornou-se uma questão de tempo até à sua disseminação ser geral, verificando-se em 1885 que os serviços de telefonia encontravam-se acessíveis à maioria dos americanos [14]. No entanto, à medida que a disseminação de telefones se tornava massiva, tornava-se também evidente que a sua utilização era limitada a locais onde se conseguisse chegar com cabos, tornando a possibilidade de comunicação limitada à capacidade de criar infraestruturas para comunicar.

Tendo em conta as limitações das redes de telefonia fixas, o físico Guglielo Marconi apresentou em 1896 [18] uma alternativa viável para os problemas impostos às redes de telecomunicações, o uso de comunicações rádio.

A vantagem associada com a propagação da informação por meio de ondas rádio encontrava-se associada com a eliminação da cablagem existente entre o terminal e o elemento recetor do sinal. Desta forma tornava-se possível eliminar a dependência física e cablada de canais de comunicação entre o utilizador e a rede. O aparecimento do paradigma rádio nas redes de telecomunicações apresentou-se essencial para o despoletar de novas soluções e possibilidades de comunicação, permitindo comunicações transatlânticas e posteriormente o desenvolvimento de terminais móveis.

Estando o ser humano habilitado com a tecnologia que lhe permite comunicar com qualquer ponto do mundo, desde que esse possua telefone, torna-se percetível que o passo seguinte na evolução das redes de telecomunicações passaria pela mobilidade do terminal. Contudo, existia um conjunto de problemáticas associadas com requisitos de mobilidade e portabilidade, como a autonomia do terminal, o peso do terminal e o modo de conectividade. O dispositivo teria de ter capacidade para a realizar uma chamada com duração considerável sem que houvesse perda de ligação devido à falta de autonomia, tendo em simultâneo de manter a ligação à rede. Estas

características tinham de ser suportadas sobre um terminal que pudesse ser transportável tanto a nível de tamanho como de peso.

O problema associado à autonomia não era de fácil resolução, embora as baterias se apresentassem como capazes de suportar o gasto energético associado ao telemóvel, estas tinham de ser de grandes dimensões tornando impraticável usar o telemóvel como um dispositivo verdadeiramente móvel, estando pois os primeiros terminais móveis associados a veículos.

A questão da conectividade era endereçada pela utilização de redes celulares, que por definição são redes de comunicação rádio que disponibilizam um subconjunto de frequências numa localização delimitada [19]. Estas redes possuem no mínimo um ponto de receção das comunicações, denominado de estação base. Neste tipo de redes, cada célula possui um subconjunto de frequências diferentes das células adjacentes de forma a existir uma menor interferência entre comunicações. Este facto permite garantir a reutilização das frequências em locais distantes e aumentar o número de utilizadores suportados.

Tirando proveito dos conceitos das redes celulares apresentadas por Douglas H. Ring e W. Rae Young [20] para a comunicação em veículos, Martin Cooper investigador da Motorola, realizou a primeira chamada de um telemóvel em 1973 [21]. Sendo que, após esta foi necessário esperar cerca de dez anos até ao lançamento comercial da primeira rede de telecomunicações sem fios [22], dando início a uma nova era da evolução das redes de telecomunicações.

Com o aparecimento dos telemóveis, surgiu um novo subconjunto das redes de telecomunicações designado por redes móveis. Sendo até à data definidas quatro gerações distintas.

A primeira geração de redes móveis surgiu por volta do início dos anos 80, destacando-se as redes Advanced Mobile Phone System (AMPS) e Total Access Communications System (TACS). Estas redes analógicas apresentavam um modelo de multiplexagem Frequency-Division Multiple Access (FDMA) [22], caracterizado pela sua simplicidade, não sendo contudo um modelo muito eficiente na utilização das frequências. Ao nível de características, as redes de primeira geração apresentavam já funções avançadas, tais como capacidade de realização de funções de *handover* e funções de *roaming* [23]. Contudo, devido à falta de normalização entre os diferentes sistemas implementados, não era possível efetuar comunicações noutros países.

A segunda geração de redes móveis surgiu de um processo de revolução. A passagem entre gerações caracteriza-se desta forma devido a não se verificar um processo de migração do sistema existente para um sistema de maiores capacidades [24], tendo para a segunda geração sido criada uma rede distinta não reaproveitando a infraestrutura das redes de primeira geração. De

entre as características das redes de segunda geração, destaca-se o facto de serem redes digitais, possuírem um maior grau de normalização, acautelarem ligações de dados e disponibilizarem um espectro mais alargado de funções. No que respeita às questões de multiplexagem, é abandonado o modelo FDMA passando a ser usados modelos Time-Division Multiple Access (TDMA) ou Code-Division Multiple Access (CDMA) [22]. De entre as propostas de redes de segunda geração destacam-se as redes Global System for Mobile Communications (GSM) [25] e as redes Interim Standard 95 (IS-95) [26].

As redes GSM tiveram o seu desenvolvimento na Europa, sendo o esforço de normalização iniciado em 1991 [22]. O modelo GSM apresenta-se tendo como princípio base a uniformização dos requisitos associados às redes de segunda geração, tentando diminuir custos relacionados com o desenvolvimento de terminais, redes de acesso e manutenção de equipamentos. Em termos funcionais, as redes GSM operam usando o modelo de multiplexagem TDMA, dividindo cada banda de 200 kHz em 8 blocos temporais [27], sendo estes alocados aos utilizadores aquando da utilização da rede. Seguido o paradigma de comutação de circuitos, as redes de segunda geração têm a necessidade de estabelecer o percurso entre os dois intervenientes na sessão antes de ser possível efetuar a chamada.

As redes IS-95, designadas de redes cdmaOne surgiram no início da década de 90 [26], diferenciando-se das redes GSM pelo modelo de multiplexagem seguido, neste caso CDMA. Estas redes tiveram maior repercussão nos Estados Unidos e, ao contrário das redes GSM, não se apresentavam como sendo a única solução aceite. Relativamente ao modelo de multiplexagem usado, o modelo CDMA apresenta-se bastante diferente do modelo seguido nas redes GSM. No modelo CDMA é utilizado uma banda de 1.23 MHz [22], sendo usada em todas as células por todos os utilizadores. Cada utilizador é associado a uma chave de código único, sendo a chave usada para a codificação e descodificação do sinal associado à chamada.

A possibilidade de transferir dados é uma característica presente em ambas as redes de segunda geração. No entanto, com o passar do tempo, as larguras de banda disponibilizadas apresentaram-se insuficientes para as necessidades e requisitos dos utilizadores, verificando-se a necessidade de aumentar as capacidades de dados. Desta necessidade surgiram as redes denominadas de 2.5G, sendo a rede General Packet Radio Service (GPRS) [22] uma das primeiras normalizações, que se caracteriza como sendo uma rede de 2G com suporte de protocolos de comutação de dados tendo subsequentemente, maior capacidade de largura de banda.

Embora as redes 2.5G se apresentassem como sendo uma evolução essencial na transferência de dados nas redes móveis, ainda possuíam limitações ao nível da transferência de dados.

As ligações de dados entre o terminal e a estação base ainda seguiam o modelo de circuitos, evidenciando-se perdas de desempenho neste fragmento da rede. Para além das limitações associadas ao desempenho das redes, verificam-se diferentes normalizações em diferentes pontos do mundo, dificultando a implementação de um serviço capaz de ser independente da plataforma.

As problemáticas associadas com as redes de 2.5G despoletaram o aparecimento das redes 3G. De entre as diferentes implementações de redes 3G, destacam-se as redes Universal Mobile Telecommunication System (UMTS) e as redes CDMA2000, apresentando ambas os mesmos princípios base (suporte de um vasto leque de serviços e maiores larguras de banda) [28]. Ao contrário das redes de 2G, as redes 3G não se apresentavam como uma rede criada de raiz, mas sim como uma evolução das infraestruturas já existentes para dar suporte aos novos requisitos. A decisão de reutilização das redes de segunda geração entrou em conta com os fatores económicos. Os operadores que necessitaram de desenvolver uma rede de segunda geração pretendiam continuar a tirar receitas da rede de forma a rentabilizar as infraestruturas, sendo para os operadores impensável a criação de uma nova rede de raiz. Contudo, em muitos locais do globo a implementação das redes 3G foi adiada até aos dias de hoje. Esta resistência à mudança tinha dois motivos principais, o preço das licenças [29] e a necessidade de criar uma infraestrutura de raiz pelo facto das frequências das redes de segunda geração e terceira geração não serem na mesma gama em todos os locais do globo.

As limitações económicas colocam entraves na evolução das redes de telecomunicações contudo, a procura por maiores capacidades, acesso a ligações *Broadband* e a disponibilização de serviços por parte da rede para os utilizadores, impulsionaram o desenvolvimento das redes de 4G. Com o desenvolvimento tecnológico verificado tanto ao nível de terminais como da rede tornou-se imperativo desenvolver uma *framework* capaz de dar suporte a todos os serviços disponibilizados pelas redes legadas independentemente do terminal, atingindo desta forma níveis de ubiquidade até à data inexistentes.

De forma a atingir os níveis de qualidade desejados, os investigadores começaram a desenvolver uma arquitetura capaz de operar sobre diversas redes de telecomunicações de modo a harmonizar os serviços. Da necessidade de conciliar as diferentes redes de telecomunicações decidiu-se abandonar o paradigma de circuitos nas redes de voz, passando a usar o paradigma dea comutação de dados no *core* das redes 4G. A passagem do paradigma de circuitos para dados justifica-se pelo volume de tráfego que é gerado na atualidade, verificando-se ao longo dos últimos anos o aumento do tráfego de dados em detrimento do tráfego de voz [30].

A mudança de paradigma e as melhorias efetivas da mudança para uma rede NGN, apresentam-

se essenciais para o desenvolvimento tecnológico e cultural de uma nação, sendo a evolução nas redes de telecomunicação apontados pela ITU-T como um ponto fundamental para o desenvolvimento e alcance dos objetivos do milénio, '*ICTs and broadband networks have become vital national infrastructure— similar to transport, energy and water networks — but with an impact that promises to be even more powerful and far-reaching. These key enhancements in wireless broadband can drive social and economic development, and accelerate progress towards achieving the United Nations' Millennium Development Goals, or MDGs.*' [31].

2.2 Arquitetura das Redes de Próxima Geração

2.2.1 ITU-T NGN

A iniciativa de desenvolvimento das NGN foi despoletada pela ITU-T em finais de 2003 [32], definindo os princípios sobre os quais iriam assentar às redes NGN. Segundo a ITU-T, as NGN são definidas como: '*A packet-based network able to provide telecommuncation services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies It enables unfettered access for users to networks and to competing service providers and/or services of their choice. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.*' [5].

A definição da ITU-T apresentava de forma muito concreta quais as ideias e os requisitos que se pretendiam por parte das redes NGN, não se verificando ao longo de todo o processo de normalização discussão sobre a componente técnica subjacente às NGN. Os princípios apresentados pela ITU-T descrevem as NGN como redes capazes de fazer a convergência entre as redes fixas e móveis. Neste sentido, abrangem todos os serviços disponibilizados pelas redes legadas e encontram-se conceptualmente preparadas para receber os novos serviços desenvolvidos, sendo todos os serviços suportados sobre uma infraestrutura única.

Todos os princípios definidos para as NGN são atingidos tirando proveito dos conhecimentos obtidos das redes de dados. Mudando o paradigma usado até à altura para a voz de circuitos para dados, conseguiu-se aumentar a flexibilidade de manuseamento da informação permitindo uma massificação dos serviços. O facto dos serviços de voz serem disponibilizados sobre paradigmas de dados, mostra uma reorganização interessante nas redes de telecomunicações. Inicialmente os dados eram passados sobre os canais de voz como se de voz se tratasse, na atualidade a voz é

tratada na redes de dados como se de simples pacotes de rede se tratassem.

A modificação do paradigma presente nas redes de telecomunicações só foi possível pela capacidade das NGN diferenciarem e darem prioridades distintas aos diferentes tipos de tráfego, tornando-se possível realizar tratamentos diferenciados consoante as características do tráfego. Este processo é conseguido através de tecnologias de QoS [33] capazes de definir diferentes prioridades consoante os requisitos dos serviços adjacentes. Deste modo, disponibilizam-se serviços das redes legadas sobre ligações de dados garantindo a qualidade da sessão.

A intenção de suportar os diferentes serviços, das diferentes redes legadas, leva à necessidade de separação da arquitetura para tornar os serviços independentes. No entanto, os serviços só conseguem ser independentes das plataformas de transporte se não estiverem diretamente associados às funções de transporte, sendo necessário desenvolver a rede em módulos independentes. Através de uma arquitetura modular, a rede torna-se capaz de suportar diferentes terminais com diferentes capacidades, dando sempre o mesmo conjunto de serviços ao utilizador, criando desta forma um sistema verdadeiramente ubíquo e unificador. A capacidade de unificar os serviços é suportada pela estrutura arquitetural apresentada pelas redes NGN (Figura 2.1).

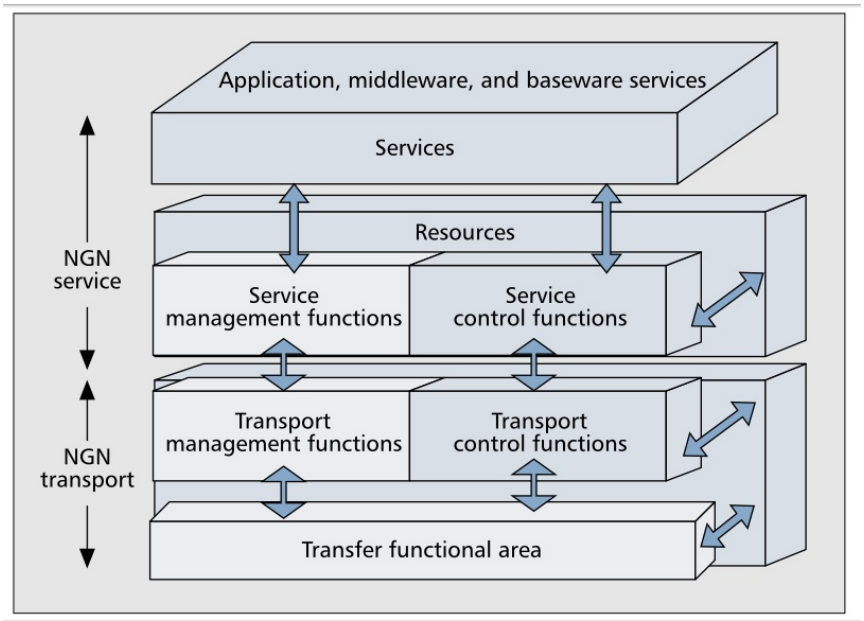


Figura 2.1: Modelo funcional definido pela ITU-T [1]

Na arquitetura definida pela ITU-T é evidenciada a demarcação entre o transporte e os serviços. Ao nível do transporte são tratadas as questões relacionadas com toda a conetividade dos diferentes elementos, desde a ligação dos terminais à rede até à conetividade entre os elemen-

tos de rede, sendo também, ao nível do transporte, endereçadas as necessidades de QoS fim-a-fim desejáveis nas redes NGN. Ao nível do serviço, são tratados todos os serviços disponibilizados nas redes NGN, tanto os serviços baseados em sessões, como os não baseados em sessões [1]. Os serviços baseados em sessões caracterizam-se como serviços disponibilizados sobre uma sessão, podendo ser serviços de videoconferência, voz sobre IP, entre outros. Já os serviços não baseados em sessões caracterizam-se como serviços não associados a sessões, sendo exemplo deste tipo de serviços os serviços de *streaming* e acesso à rede [1].

2.2.2 ETSI TISPAN

Ao mesmo tempo do desenvolvimento por parte da ITU-T dos princípios e ideias por detrás das NGNs, o instituto europeu de normalizações no ramo das telecomunicações (ETSI) iniciou esforços para a especificação de uma arquitetura da rede NGN, levado a cabo pela TISPAN.

Em 2003, a TISPAN iniciava os trabalhos de especificação da arquitetura NGN [34], tendo lançado a primeira versão em 2005 [35]. Esta assumia um papel fundamental para o desenvolvimento prático das NGN pois apresentava-se como uma definição normalizada e robusta, capaz de ser seguida por diferentes fabricantes para o desenvolvimento de soluções NGN. Em termos funcionais, a TISPAN apresentava as NGN como redes completamente IP, sendo a escolha do IP uma escolha lógica tendo em conta toda a disseminação e diversidade de normas associadas ao protocolo. Para além do uso do protocolo IP, outro dos grandes trunfos das NGN está associado com a modularidade da arquitetura definida pela TISPAN. A normalização apresentada pela TISPAN compreende as NGN como redes constituídas por subsistemas, tendo cada um funções específicas. De entre os subsistemas definidos destaca-se o IMS, sendo este responsável pela gestão dos serviços baseados em sessão (ex. serviços disponibilizados sobre SIP).

A segunda versão das NGN é lançada em 2008 [36], dando ênfase aos serviços de IPTV sobre as redes IMS e redes não IMS e à disponibilização de serviços *triple-play* e *quadruple-play*. Até à data ainda não foi lançada a terceira versão, existindo contudo esforços contínuos para melhorar as normas em diversas áreas relacionadas com o tratamento de QoS, segurança, entre outras.

2.3 IP Multimedia Subsystem

A rede IMS é um elemento essencial das redes NGN, apresentando-se como o subsistema responsável pela operacionalização dos serviços disponibilizados sobre sessões. A primeira definição da rede IMS surge na versão 5 do 3GPP [37], sendo a rede descrita como uma arquitetura funcional capaz de suportar um conjunto de serviços, fazendo convergência sobre diferentes acessos.

Para os fornecedores de serviço, as redes IMS apresentam-se como uma oportunidade única para a mudança do modelo de negócio, alargando os serviços disponibilizados aos utilizadores e "libertando-se" da conveniência obtida pelos serviços de voz sobre circuitos. Grande parte do volume de negócio dos operadores advém da taxação dos serviços de voz, sendo este também o principal serviço disponibilizado pelas redes. Contudo, com o crescente aumento das capacidades das ligações de dados e devido à melhoria das aplicações de voz sobre IP, tem-se verificado uma diminuição do mercado de voz sobre as redes de circuitos, tornando-se prática comum o uso de aplicações VoIP para realização de chamadas. Deste modo, os operadores retiram receitas somente da taxação da ligação e não do serviço, assumindo assim um papel de *dumb pipe* [38], transmitindo os dados entre a rede e o terminal.

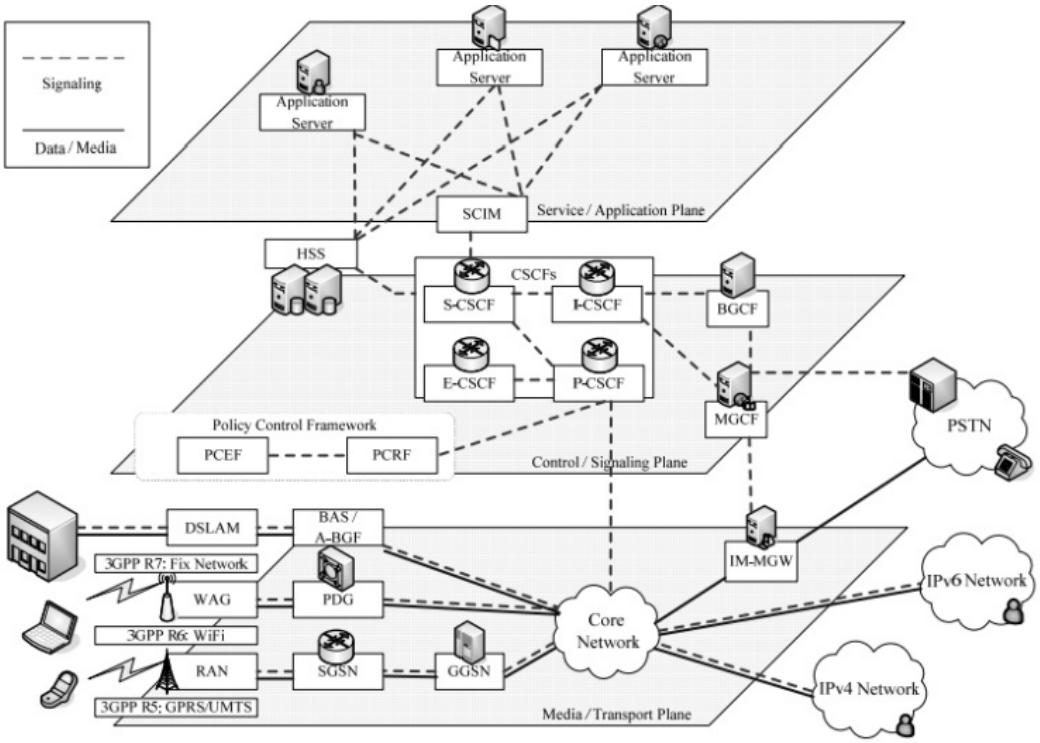


Figura 2.2: Arquitetura estratificada IMS [2]

As redes IMS contrariam as tendências presentes no mercado das telecomunicações, permitindo aos operadores disponibilizar um conjunto de serviços multimédia com mecanismos de taxação adequados. Em termos dos serviços disponibilizados, na sua maioria são serviços já existentes sobre as redes de dados. Contudo, os serviços das redes IMS garantem o cumprimento dos requisitos de QoS, tendo preocupações com segurança e disponibilizando o serviço de forma independente do terminal.

A capacidade de aglutinação dos diferentes serviços das diferentes redes torna-se possível pela utilização de um modelo arquitetural dividido em camadas como o apresentado na Figura 2.2.

A nível arquitetural o IMS apresenta-se com três camadas distintas (transporte, controlo e serviço), tendo cada uma funções diferenciadas interagindo entre si sobre interfaces normalizadas. Através da estratificação do modelo arquitetural, as redes IMS conseguem garantir a convergência de redes e terminais sobre a mesma infraestrutura, apresentando-se o IMS como um elemento unificador de serviços.

A utilização do modelo em camadas constituído por diferentes módulos simplifica a junção de novos elementos na rede e adição de novos serviços, tornando possível o crescimento consoante as necessidades. A adição de serviços de forma mais célere ocorre porque a rede não possui serviços normalizados, mas sim funcionalidades [39]. Como apresentado na Figura 2.3, ao contrário do modelo de integração vertical seguido nas redes legadas, as redes IMS seguem um modelo horizontal existindo preocupações de reutilização de funções comuns. Deste modo torna-se possível desenvolver serviços independentes das funções de mais baixo nível, aumentando a qualidade dos serviços prestados e disponibilizando estes num menor espaço de tempo.

Relativamente aos serviços disponibilizados, as vantagens destes não se encontram diretamente associadas com a inovação dos serviços prestados (muitos dos serviços já são disponibilizados na Internet), mas sim devido aos serviços possuírem as características intrínsecas das redes IMS. De todas as características da rede pode-se destacar a capacidade de garantir tratamento diferenciado consoante os requisitos de QoS do serviço, a garantia do serviço independentemente da rede de acesso, a interoperabilidade com redes legadas e a disponibilização de serviço indistintamente da localização [39].

O controlo da qualidade do serviço apresenta um papel fundamental em toda a arquitetura IMS. Tal verifica-se pois, ao contrário dos serviços prestados sobre redes legadas, os serviços das redes IMS operam sobre pacotes, sendo necessário diferenciar tráfego prioritário (ex. voz) de modo a, no mínimo, garantir que a qualidade disponibilizada nos serviços IMS é equivalente

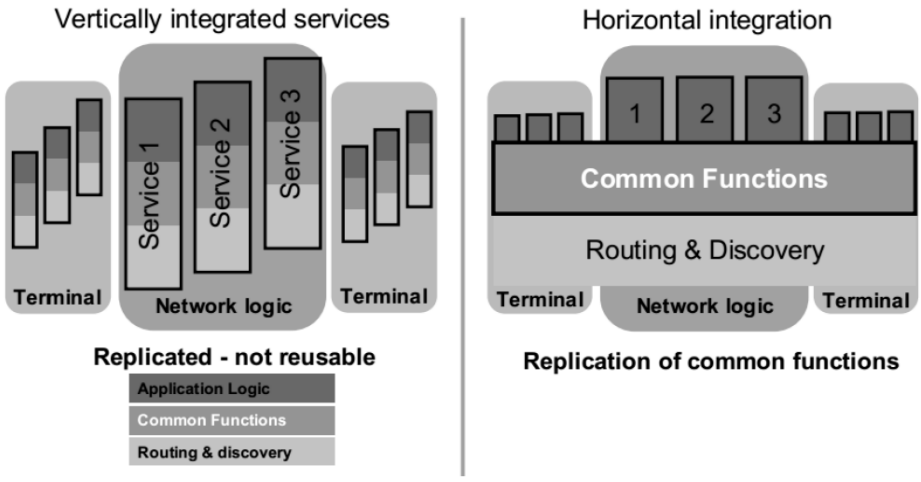


Figura 2.3: Compatativo entre arquitetura de integração de serviços das redes legadas e das redes IMS [3]

à disponibilizada nas redes legadas.

No que respeita à interoperabilidade, as redes IMS necessitam de garantir acesso às redes legadas de telefonia e de dados. A necessidade de possuir conetividade com as redes legadas verifica-se pois, embora estas últimas estejam progressivamente a entrar em desuso, a sua presença no cotidiano das pessoas irá manter-se durante um período longo de anos. Para além da necessidade de garantir suporte às redes legadas a interoperabilidade acarreta vantagens relativamente ao número de utilizadores na rede. A capacidade dos utilizadores em redes IMS comunicarem com utilizadores em redes legadas torna os serviços mais apetecíveis, aumentando a sua utilização.

Em termos de disponibilização de serviços, as vantagens dos serviços IMS, em detrimento de serviços similares disponibilizados na Internet, encontram-se associadas à capacidade dos serviços IMS disponibilizados pelo operador garantirem o serviço independentemente do terminal e do local. Desta forma é possível aceder aos mesmos serviços em terminais distintos e localizações distintas (*roaming*), tornando o sistema verdadeiramente ubíquo indo de encontro ao espectável pelo utilizador.

2.3.1 Arquitetura IMS

A arquitetura IMS apresenta-se como um arquitetura evoluída capaz de suportar um grande leque de serviços. Contudo, no panorama das NGN apresenta-se como um subsistema usado com o

intuito de gerir todo o tipo de serviços baseados em sessões presentes na rede.

Na arquitetura das NGN é apresentado o IMS como Commom IMS. Embora o IMS tenha sido desenvolvido pelo 3GPP, o seu sucesso fez com que fosse usado por diferentes entidades de normalização para o tratamento de serviços baseados em sessões. O facto de o IMS estar associado a entidades como a TISPAN, o 3GPP2 [40] e CableLabs [41] levou à existência de diferentes versões do IMS consoante os requisitos das entidades de normalização. De forma a colmatar as problemáticas associadas à existência de diferentes normalizações para o componente IMS, as entidades de normalização acordaram que a manutenção do IMS iria estar a cargo do 3GPP, obtendo o 3GGP requisitos das diferentes entidades de normalização.

Em termos funcionais, a arquitetura IMS possui todos os elementos de Core Network (CN) necessários para o fornecimento de serviços multimédia baseados em SIP [8]. A disponibilização dos serviços é realizável tirando partido de uma arquitetura estratificada, possuidora de elementos com funções de sinalização e *bearer* dispersos pelas diferentes camadas. Através do modelo usado simplifica-se a tarefa de convergência dos serviços de voz, vídeo e dados, tirando melhor partido das sinergias entre a Internet e as redes de telecomunicações.

No que respeita à implementação física das funcionalidades do IMS, estas podem estar centralizadas num elemento ou dispersas por elementos individuais, não fazendo parte do cariz da normalização indicar de que forma os elementos devem estar distribuídos a nível físico. Esta liberdade de implementação deve-se ao facto da normalização definir elementos funcionais que não necessitam verdadeiramente de corresponder a um nó físico [39]. Os equipamentos físicos que implementam as entidades lógicas são interligados usando interfaces normalizadas, permitindo desta forma, promover a interoperabilidade entre diferentes fabricantes, diminuir o custo e melhorar a qualidade da solução na sua globalidade. Seguindo estes princípios, o operador é livre de comprar equipamentos de diferentes fornecedores e colocar estes no mesmo ambiente a comunicar.

De entre os elementos presentes na arquitetura IMS destacam-se alguns dos apresentados na Figura 2.4, sendo as suas funcionalidades descritas de seguida.

O **Call Session Control Function** (CSCF) é um componente lógico fulcral na arquitetura IMS. As entidades CSCF operam como SIP Servers estando diretamente associadas com grande parte das funcionalidades referentes à sinalização na arquitetura IMS (funções de alocação de recursos, encaminhamento e autenticação) [39], podendo apresentarem-se na rede na forma de quatro entidades lógicas distintas [42].

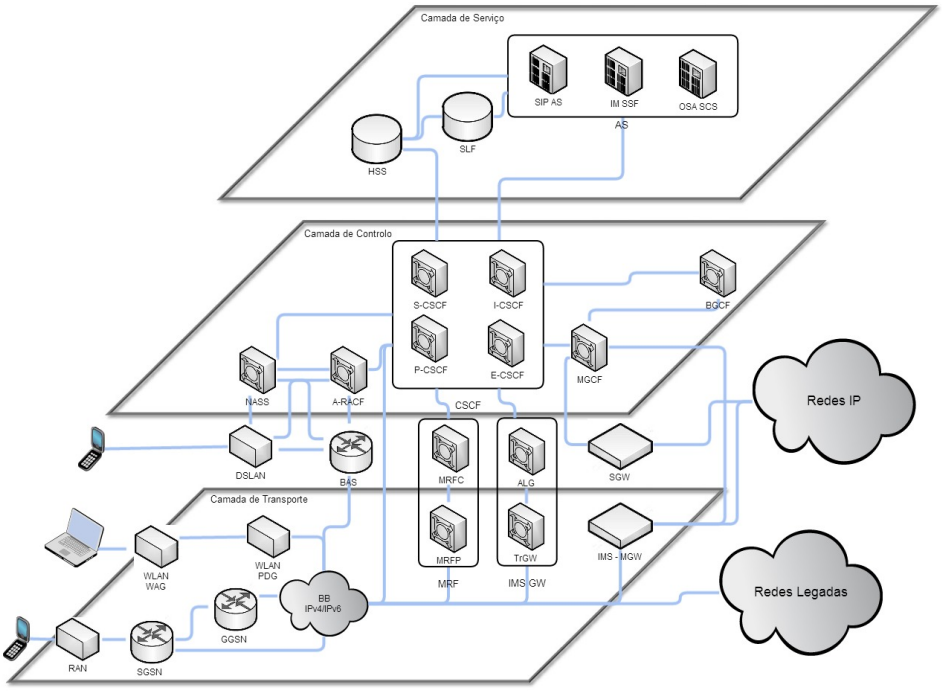


Figura 2.4: Arquitetura da rede IMS

O Proxy-CSCF (P-CSCF) é o primeiro elemento que o User Entity (UE) contacta quando pretende usar a rede IMS independentemente de se encontrar na rede do seu operador (*home network*) ou em situação de *roaming* (*visited network*). Ao nível operacional, o P-CSCF atua como um SIP Proxy Server fazendo o encaminhamento de mensagens SIP de e para o UE. Em termos funcionais, opera de modo stateful [9], endereça questões de autenticação e autorização, realiza validações sobre o protocolo de sinalização e possui funções de compressão/descompressão de mensagens de sinalização [39].

O P-CSCF é responsáveis pela criação de uma ligação confiável entre o UE e o P-CSCF, sendo feita sobre IP Security Protocol (IPsec) após autenticação do utilizador por parte da rede. O facto do utilizador se apresentar perante o P-CSCF como um utilizador seguro, permite ao P-CSCF assegurar aos restantes elementos da rede a confiabilidade do UE.

A validação das mensagens SIP apresenta-se como um ponto crucial para o bom funcionamento da rede. Com frequência, mesmo sem a existência de um comportamento deliberado de alteração das mensagens, verifica-se que os terminais enviam mensagens que não se encontram de acordo com o esperado pela rede. De forma a evitar mensagens mal formatadas enviadas de ou para elementos errados. O P-CSCF analisa as mensagens recebidas tendo em conta diversos

cabeçalhos como o From, Contact, To, Route, entre outros.

O Interrogating-CSCF (I-CSCF) é o ponto de entrada do UE na rede do seu operador quando se encontra em processo de *roaming*, estando o I-CSCF responsável pelo encaminhamento das mensagens para o Serving-CSCF (S-CSCF). Aquando da receção do pedido de registo do UE, o I-CSCF contacta o HSS para saber qual o S-CSCF alocado ao utilizador. O HSS analisa a informação do utilizador, mais precisamente o perfil do utilizador, verificando qual o S-CSCF associado ao utilizador. Após escolha do S-CSCF, o HSS contacta o I-CSCF sendo de seguida da responsabilidade do I-CSCF encaminhar o pedido para o S-CSCF indicado. Concluído o processo de encaminhamento da mensagem para o S-CSCF, o I-CSCF não necessita de se encontrar mais presente no fluxo das restantes mensagens. Contudo, ocasionalmente o I-CSCF mantem-se no fluxo de forma a encobrir a topologia das redes interligadas, sendo este mecanismo denominado de THIG (Topology Hiding Inter-network Gateway) [43].

O S-CSCF é o nó central da entidade CSCF, opera como um SIP Proxy Server, podendo possuir funções de SIP Registrar. Em termos de funcionais, o S-CSCF tem funções de encaminhamento de sessões multimédia, controlo de sessão dos terminais, gestão do estado das ligações e iteração com as plataformas de serviços. Ao nível do encaminhamento o S-CSCF encontra-se responsável por encaminhar as sessões para a camada aplicacional, para o UE ou para uma rede externa. No que diz respeito ao controlo das sessões, este é feito em conjunto com o HSS. Quando um pedido chega ao S-CSCF é analisado tendo em conta o Initial Filter Criteria (IFC) associado ao UE, obtendo desta forma informação sobre os ASs a serem despoletados. Relativamente à manutenção e gestão das sessões, o S-CSCF apresenta-se como o componente responsável por indicar a inicialização e término dos diferentes serviços, auxiliando em questões de taxação, ocupação de equipamentos, entre outras.

O Emergency-CSCF (E-CSCF) é a entidade do CSCF responsável por endereçar serviços de emergência [44], tendo como objetivo o tratamento eficiente dos aspetos relacionados com o redireccionamento de chamadas diretamente para as Public Safety Answering Point (PSAP). Ao nível funcional, o E-CSCF requer que seja possível realizar chamadas independentemente do acesso e do utilizador, sendo tratadas como prioritárias. Ao nível arquitetural, o E-CSCF obtém informação do Location Retrieval Function (LRF) sobre a localização do utilizador e dos PSAPs auxiliando o processo de estabelecimento das chamadas.

O **Home Subscriber Server** (HSS) é a base de dados principal nas arquiteturas de redes IMS, onde reside informação relativa aos utilizadores, serviços disponibilizados e funcionalidades da rede. Na distribuição da informação, o HSS pode operar de forma centralizada, existindo uma

entidade central com a totalidade da informação, ou pode operar de forma descentralizada, existindo mais que uma entidade HSS, cada uma com um subconjunto de utilizadores. A decisão de utilizar só uma entidade HSS ou usar várias entidades depende do número de utilizadores na rede, da capacidade dos HSS e da própria organização da rede. Ao nível de serviços, verifica-se que a rede IMS só consegue realizar funções de autenticação, localização e taxação através da utilização da informação obtida pelo HSS. Em termos de informação armazenada, o HSS possui informação de identificação dos utilizadores, numeração, informação de segurança, informação de localização e informação de serviços associados aos utilizadores através dos perfis de utilizadores [45].

O **Transition Gateway** (TrGW) opera sobre o plano de *media* funcionando como um Network Address Port Translator-Protocol Translator (NATPT/NAPT-PT) [39]. Em termos funcionais, permite o mapeamento de mensagens IPv4 em IPv6 e vice-versa, permitindo a utilização de diferentes protocolos IP nas redes de acesso.

O **IMS Application Layer Gateway** (IMS-ALG) opera sobre o plano de sinalização apresentando-se em termos funcionais similar ao TrGW. Ao nível operacional, o IMS-ALG consegue fazer o mapeamento entre IPv6 e IPv4 operando como SIP B2BUA [39]. Neste modo a entidade possui uma conexão em cada rede IP permitindo assim a comunicação entre ambas.

O **Media Gateway Control Function** (MGCF) é um elemento fronteira das redes IMS com funções de interligação com as redes Public Switched Telephone Network (PSTN). Em termos técnicos, o MGCF é responsável por fazer o controlo da correlação entre as mensagens SIP e ISDN User Part (ISUP) ou Bearer-Independent Call Control (BICC), de modo a permitir a comunicação entre as diferentes redes [46]. Ao nível do *media*, o MGCF possui funções de reserva de recursos a múltiplas e heterogéneas Media Gateways (MGW).

O **Signaling Gateway** (SGW) é responsável por fazer a conversão de mensagens de sinalização entre as redes IMS e redes PSTN. As conversões realizadas são elaboradas nas camadas protocolares inferiores, sendo as conversões realizadas de ISUP sobre MTP para ISUP sobre SCTP/IP ou vice-versa [39].

O **IMS Media Gateway** (IMS-MGW) apresenta funções similares às presentes ao nível de sinalização pelo SGW, sendo neste caso aplicadas ao plano de *media*. Nas redes PSTN o *media* é trocado usando o método Pulse Code Modulation (PCM) enquanto que nas redes IMS é trocado sobre RTP, sendo necessário realizar conversões de forma a permitir a comunicação [39].

O **Media Resource Function Controller** (MRFC) encontra-se responsável por fazer a alo-

cação das *streams* de *media* para a entidade Media Resource Function Processor. Ao nível funcional, o MRFC obtém informação da rede (ex. AS) e aloca recursos para diferentes serviços de transcodificação, serviços de áudio-conferência, entre outros.

O **Media Resource Function Processor** (MRFP), apresenta-se como o elemento responsável pela implementação de funções de processamento de *media* [47] que garantem suporte às funções de mistura de *streams*, análise de áudio, geração de anúncios, entre outras.

O **Breakout Gateway Control Function** (BGCF), é responsável pela escolha do elemento da rede PSTN ou IMS para o onde devem ser encaminhadas as mensagens SIP com destino a um utilizador da rede PSTN. A decisão de encaminhamento é tomada tendo em conta o conteúdo das mensagens de sinalização ou através da consulta de entidades externas [42] (HSS, bases de dados ENUM, servidores Domain Name System (DNS), entre outras).

O **Application Server** (AS) é responsável pela disponibilização de um vasto conjunto de serviços nas redes IMS. Tipicamente, as redes IMS possuem um conjunto de ASs, cada um com um conjunto de serviços a disponibilizar. Estas entidades podem ser de diferentes tipos, podendo funcionar como SIP Proxy Server, SIP User Agent, SIP Redirect Server ou SIP Back-to-Back User Agent, estando a variação do modo de funcionamento associada ao papel que o AS possui na sessão. Quando o AS presta serviços como um terminal (ex. serviços de alertas), este funciona como um User Agent. Quando presta serviços do tipo conferência funciona como um SIP B2BUA e quando é usado para redirecionar chamadas para terminais opera como um SIP Redirect Server.

Em relação ao tipo, existem três tipos distintos de ASs [39], os SIP ASs, os OSA-SCS e os IM-SSF. Os SIP ASs apresentam-se como os ASs preferencialmente usados nas redes IMS, sendo estes os elementos sobre os quais se pretende desenvolver os serviços IMS. Os Open Service Access-Service Capability Servers (OSA-SCS) são ASs que providenciam acesso a serviços da OSA *framework*. Os IP Multimedia Service Switching Function (IM-SSF) são ASs usados para a reutilização de serviços prestados por plataformas como o Customized Applications for Mobile network Enhanced Logic (CAMEL). Do ponto de vista do S-CSCF os diferentes ASs são apresentados como idênticos independentemente do tipo ou da sua localização, podendo estes estarem na rede IMS do operador ou num *third-party*.

2.3.2 Protocolos IMS

A arquitetura IMS assenta o seu funcionamento em protocolos já existentes e normalizados. Esta opção foi tomada por parte do 3GPP após análise do trabalho realizado pela ETSI no desenvolvimento das redes GSM [39]. Aquando do desenvolvimento da rede de segunda geração a ETSI criou de raiz a grande maioria dos protocolos usados, reutilizando somente um conjunto restrito de protocolos anteriormente definidos pela ITU-T. Esta opção apresentou-se como a solução possível pois na altura do desenvolvimento da rede existia grande falta de protocolos com capacidade de dar resposta aos desafios apresentados pelas redes GSM. Quando o 3GPP decidiu iniciar trabalhos nas redes IMS, estudou o caso das redes GSM de forma a perceber qual seria o melhor caminho a seguir no que respeita a reutilização de protocolos. Após análise dos protocolos e possibilidades existentes o 3GPP optou pela reutilização de protocolos na arquitetura IMS. Reutilizando protocolos, o 3GPP conseguia desenvolver as redes IMS num espaço de tempo mais curto não colocando em causa a robustez da solução. Para além dos protocolos de gestão de sessões era necessário garantir funções de Autenticação, Autorização e Accounting (AAA) e as funções de transporte de dados multimédia, tendo a rede IMS três principais protocolos, SIP, Diameter e RTP.

2.3.2.1 Protocolos de Sinalização

SIP

De entre todos os protocolos presentes nas redes IMS, os protocolos de controlo de chamadas apresentam um papel chave na sinalização, assumindo papel de destaque, o protocolo SIP. Em termos históricos, o primeiro rascunho SIP foi disponibilizado em fevereiro de 1996, sendo este definido como um protocolo fim-a-fim usado para iniciar, modificar ou terminar sessões multimédia IP [48]. A primeira definição aceite do protocolo surgiu por parte da Internet Engineering Task Force (IETF) em março de 1999 na Request for Comments (RFC) 2543 [49]. Esta normalização viria a ser alvo de atualizações em 2002 na RFC 3261 [9], mantendo-se na atualidade como a RFC do protocolo. Em termos de caraterísticas, o protocolo SIP assenta o seu funcionamento no modelo pedido-resposta, possuindo um elemento como cliente e outro como servidor. No que diz respeito as mensagens trocadas, estas são enviadas entre o cliente e o servidor no formato de texto. Com o desenho do protocolo SIP pretendia-se criar um protocolo de gestão de sessões simples, modular, escalável que permitisse a mobilidade tanto das pessoas como das sessões, que fosse extensível e que se apresentasse independente do protocolo de *media* uti-

lizado [48].

A sinalização possui um papel fundamental no estabelecimento de uma sessão. Através deste processo é negociada toda a informação relativa à sessão a estabelecer, sendo necessário um processo simples e escalável de forma a causar o menor impacto possível na rede. No que diz respeito ao problema associado à escalabilidade do sistema, o protocolo SIP endereça-o separando funções por diferentes elementos definidos dentro do próprio protocolo, conferindo à rede a possibilidade de crescer consoante as necessidades dos utilizadores e do volume de tráfego.

Com a crescente mobilidade de pessoas e terminais, torna-se necessário que o utilizador seja identificado e contactado usando sempre a mesma identidade. Este critério é acautelado pelo SIP no processo de registo do utilizador, tendo o protocolo entidades específicas para o mapeamento e localização dos utilizadores.

No que respeita à capacidade de mudança, o SIP acompanha a evolução das redes de forma bastante eficaz através da utilização de extensões ao protocolo base. A RFC base do SIP tal como referido anteriormente é a RFC 3261. Contudo, nesta norma não estão contidas todas as funcionalidades do protocolo SIP, estando diversas funcionalidades identificadas em RFCs de extensões ao protocolo base. De entre as diferentes RFCs de extensões podem-se referir normalizações como, a norma do cabeçalho Refer [50], a norma responsável pelas funções de mensagens instantâneas [51] e a normalização de localização de servidores SIP [52]. Ao todo existem mais de 60 RFCs definidas para o protocolo SIP [48], representando cada uma funcionalidades ou cenários específicos associados ao protocolo.

Em termos funcionais o que se pretende com o protocolo SIP resume-se à possibilidade de criar uma sessão e a manter sem ter de saber onde os utilizadores se encontram.

A título de exemplo, a sinalização SIP trocada entre dois terminais pode corresponder às mensagens de voz trocadas entre dois utilizadores através de *walkie talkies* (Figura 2.5). Nesta conversa, o originador da chamada indica que pretende falar e pergunta se o recetor pode ouvir, caso o recetor possa estabelecer a chamada é trocada a mensagem, sendo no final terminada a comunicação. No SIP como representado na Figura 2.6, as mensagens são trocadas através de pedidos e respostas correspondendo estas a métodos SIP e respostas de status [9].

Os métodos são mensagens de pedido usadas por exemplo para efetuar registos, de-registos e realizar chamadas. Estes na RFC base são definidos como de seis tipos distintos (Tabela 2.1), sendo contudo mais quando consideradas as extensões SIP.

As mensagens de status são usadas para o envio de respostas aos métodos, podendo indicar

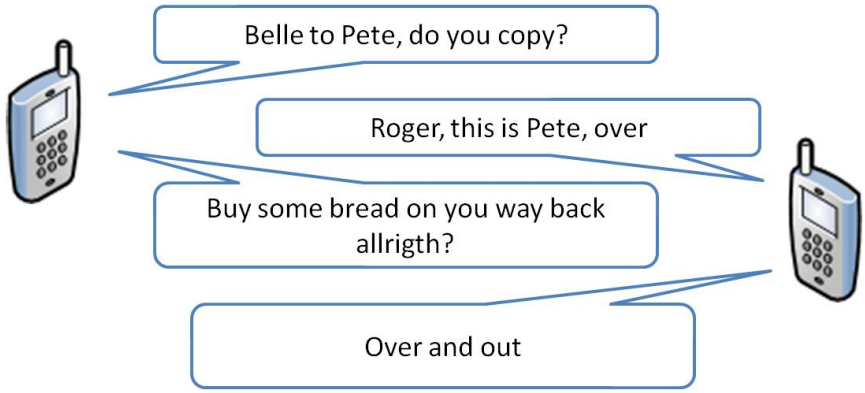


Figura 2.5: Conversa telefónica com fluxo de sinalização similar ao presente em SIP

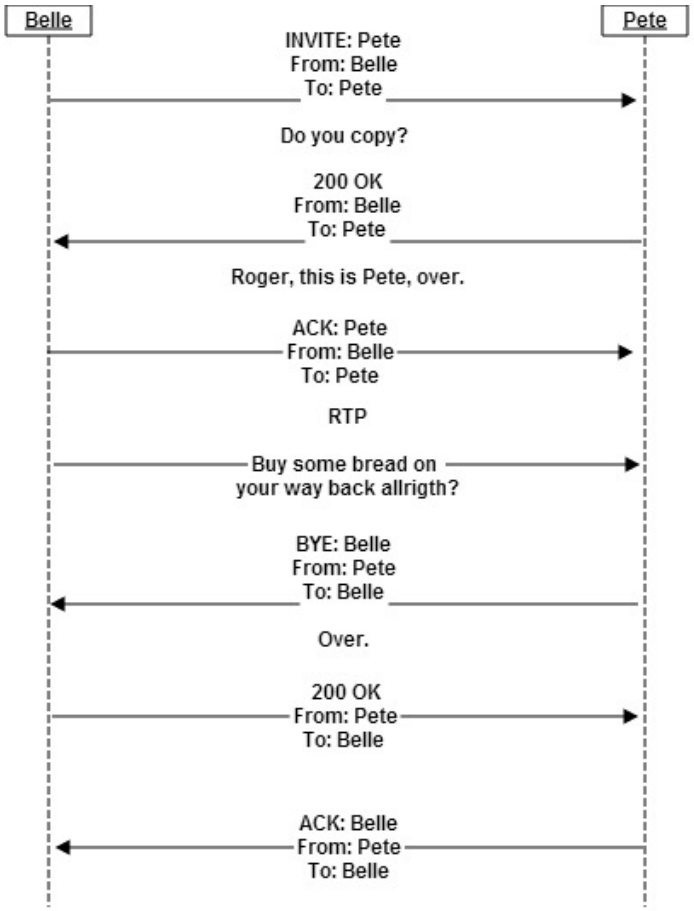


Figura 2.6: Mapeamento de um conversa num fluxo SIP

Metodos	Descrição
REGISTER	Método usado para fazer o registo e de-registo do terminal
INVITE	Método associado à iniciação de uma sessão e/ou revalidação da mesma
ACK	Método de acknowledge a um pedido, a maioria dos métodos possuem um ack associado
BYE	Método usado para colocar termo a uma sessão
CANCEL	Método usado pelo originador da sessão para o término da sessão ainda em fase de negociação
OPTIONS	Método usado para questionar um elemento sobre as suas capacidades

Tabela 2.1: Métodos associados com a RFC base do SIP

sucesso ou insucesso dos métodos e a causa associada. Em relação à sua classificação, as mensagens de status apresentam-se dívidas em seis grupos como apresentado na Tabela 2.2, tendo cada grupo um conjunto de mensagens associadas.

Respostas	Descrição
1xx	Respostas provisórias, usadas entre o envio do pedido e a obtenção da resposta final
2xx	Respostas de sucesso, tipicamente a mensagem 200 OK
3xx	Respostas de redireccionamento, indicam redireccionamentos temporários ou definitivos
4xx	Respostas de erro do cliente, usadas quando o erro foi despoletado pelo cliente (ex. Bad Request)
5xx	de erro do servidor, situações de erro despoletado pelo servidor (ex. Gateway Timeout)
6xx	Resposta de falha global, resposta despoletada quando se verifica a impossibilidade de contactar o utilizador (ex. Busy Everywhere)

Tabela 2.2: Status associados com a RFC base do SIP

Em termos estruturais, as mensagens SIP (métodos e status) possuem inicialmente a informação do pedido ou resposta associada à mensagem, de seguida um ou mais cabeçalhos SIP e por fim podem apresentar o corpo de mensagem. Relativamente aos cabeçalhos presentes nas mensagens SIP, estes podem ser de cariz obrigatório ou opcional. Os cabeçalhos obrigatórios necessitam de se encontrar presentes, sendo despoletados erros pela falta dos mesmos. Relativamente aos cabeçalhos opcionais a sua presença por norma encontra-se associada a uma função auxiliar a um determinado cenário.

Ao nível arquitetural, o protocolo SIP possui um elevado grau de escalabilidade devido à separação de funções pelos diversos elementos da rede.

Os User Agents (UAs) representam equipamentos terminais responsáveis por iniciar e receber as sessões SIP. Os dois terminais que se encontram envolvidos na sessão possuem o nome de User Agent Client (UAC) e User Agent Server (UAS). O UAC é o elemento que realiza o pedido, enquanto que o UAS é o elemento que recebe o mesmo. Por norma, ambos os elementos podem operar como UAC e UAS, tendo papéis diferentes consoante o envio ou receção dos pedidos.

Os Proxy Servers são servidores SIP com funções de encaminhamento e validação de mensagens usados ao longo da rede para encaminhar as mensagens SIP para terminais ou *proxies*. Relativamente às validações, estas têm por base a análise do conteúdo da mensagem comparativamente com o definido na gramática Augmented Backus-Naur Form (ABNF) [53] associada ao SIP, verificando se o conteúdo da mensagem se encontra de acordo com o formato expectável. O encaminhamento é realizado através da análise de cabeçalhos da própria mensagem, ou contactando entidades externas. Na própria mensagem, cabeçalhos como Router ou Via indicam como as mensagens podem ser encaminhadas. Recorrendo a elementos externos, o encaminhamento é feito através do contacto com servidores DNS, base de dados ENUM, entre outros.

Os Redirects Servers têm funções no processo de redireccionamento dos pedidos para outras entidades, quer estas sejam UA ou *proxies*. Aquando da existência de uma mudança temporária ou permanente de uma entidade, o Redirects Server fica responsável pela indicação de qual o destino atual para o qual comunicar. A nível arquitetural, os Redirect Servers possuem como principal função a libertação de carga dos *proxies*.

O Registrar Server é responsável pelo registo dos utilizadores e manutenção do registo. O processo de registo nas redes SIP é realizado através do envio de um pedido REGISTER para o Registrar Server. Quando o pedido é recebido no Registrar, este em conjunto com o Location Service verifica a validade do pedido e mapeia a identidade pública do utilizador com o seu contacto atual. Deste modo, é possível o estabelecimento de sessões com o utilizador através da sua entidade pública independentemente da sua localização física.

O Back-to-Back User Agent (B2BUA) representa a junção de dois componentes da arquitetura SIP, o UAC com o UAS [48]. Em termos normativos não existe uma definição na sobre o comportamento do componente porque, do ponto de vista técnico o B2BUA corresponde à junção de dois componentes já definidos na norma. Em termos funcionais, o B2BUA apresenta-se como um componente capaz de controlar a sessão tendo capacidade para terminar, iniciar e replicar sessões, dividir domínios, entre outras funcionalidades.

SDP

O protocolo Session Description Protocol (SDP) define o formato para descrição dos parâmetros de inicialização de troca de *media* [4]. Em termos funcionais, o protocolo apresenta um modelo pedido-resposta onde o UAC propõe a definição da sessão (indicação dos *codecs*, tipo de *media*) e o UAS responde com os parâmetros com que concorda. Em termos estruturais, as mensagens SDP possuem um conjunto de campos divididos por três secções distintas: a descrição das sessões, descrição de tempo e a descrição do *media* [4].

A descrição das sessões, apresenta informações sobre versões do protocolo, informação de identificação do utilizador e da sessão, tal como apresentado na Tabela 2.3.

Símbolo	Descrição
v	Versão do protocolo SDP
o	Informação do username, ID sessão e do ID versão
s	Nome da sessão
i	Título da sessão
u	URI de informação adicional relacionada com a sessão
e & p	Contactos dos elementos responsáveis pela sessão
c	Informação da conexão, presente por sessão ou por media description
b	Identificação da largura de banda proposta a ser utilizada

Tabela 2.3: Protocolo SDP: Informação da descrição da sessão [4]

A informação temporal do protocolo SDP é considerada obrigatória sendo usada para a identificação do início e fim da sessão. Este tipo de informação pode ser apresentada diversas vezes na mensagem consoante o número de vezes que a sessão necessita estar ativa. Em relação aos parâmetros, estes podem ser de três tipos distintos como apresetando na Tabela 2.4.

Símbolo	Descrição
t	Tempo do início e final de uma sessão
r	Número de repetições da sessão
z	Calendarização de execução da sessão

Tabela 2.4: Protocolo SDP: Informação da descrição temporal [4]

A informação de *media* indica o tipo de *media*, o protocolo subjacente e o formato do *media* (*codecs*) associado à sessão a estabelecer. Os *codecs* são negociados entre os terminais de forma a terem codificações compreensíveis entre ambos (nas situações de impossibilidade de acordarem num *codec* é necessário realizar operações de transcodificação na rede). No que respeita aos atributos usados, estes são apresentados na Tabela 2.5.

Símbolo	Descrição
m	Descritivo de um conjunto de parâmetros do media (tipo de *media*, protocolo transporte)
a	Atributos do *media*

Tabela 2.5: Protocolo SDP: Informação de media [4]

2.3.2.2 Protocolo de AAA

As funções de AAA nas redes IMS são endereçadas pelo protocolo Diameter, que se encontra definido na RFC 6733 [10], e num conjunto de extensões denominadas de Applications. Em termos evolutivos, o protocolo Diameter surge como evolução do protocolo RADIUS [54]. As principais diferenças entre os protocolos Diameter e RADIUS encontram-se associadas a questões de segurança, escalabilidade e desempenho. O Diameter ao contrário do RADIUS opera unicamente sobre protocolos de transporte confiáveis como o Transmission Control Protocol (TCP) ou Stream Control Transmission Protocol (SCTP) e tanto o cliente como o servidor devem ter suporte para IPSec. No que respeita a questões de escalabilidade, o Diameter possui maior capacidade para endereçamento de Attribute-Value Pairs (AVPs), passando de 8 para 32 bits, ficando desta forma os AVPs dispostos em conjuntos de 4 bytes, aumentando também o desempenho na análise dos cabeçalhos [55].

Ao nível funcional, o protocolo Diameter opera ponto-a-ponto entre entidades Diameter, podendo ser considerados três tipos de entidades distintas: clientes Diameter, agentes Diameter e servidores Diameter. Os clientes Diameter fazem o controlo de acessos, os agentes Diameter funcionam como *proxies* encaminhando mensagens e os servidores Diameter tratam os pedidos a nível da autenticação, autorização e *accounting*.

Em termos arquiteturais o protocolo Diameter apresenta-se constituído por um protocolo base e por um conjunto de Applications. As Applications são extensões ao protocolo base com funções próprias que interagem com o Diameter base como apresentando na Figura 2.7. De entre as diferentes extensões podem-se referir a Diameter Credit-Control Application usada para efeitos de taxação [56], a Diameter Network Access Server Application responsável por permitir aos servidores Diameter conseguir endereçar questões de AAA para nós móveis [57] e Diameter Session Initiation Protocol Application usada em conjunto com SIP para permitir a servidores SIP fazer pedidos a servidores Diameter [58].

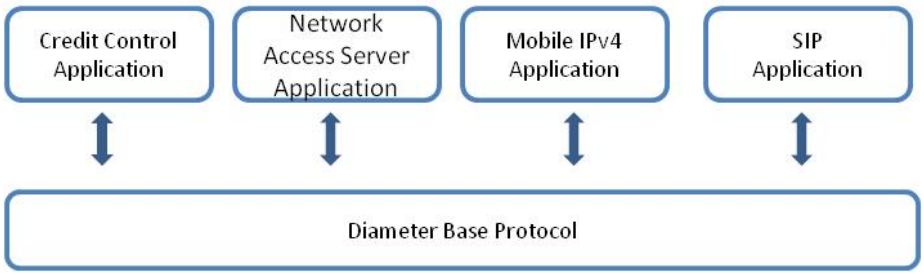


Figura 2.7: Protocolo Diameter base e aplicações

2.3.2.3 RTP

O principal protocolo de transporte de *media* das redes IMS é o Real-time Transport Protocol (RTP) [11], estando responsável pela entrega de tráfego em tempo real. A nível funcional, o protocolo segue um modelo fim-a-fim normalmente sobre o protocolo User Datagram Protocol (UDP), encontrando-se associado a um vasto leque de serviços de *streaming* de áudio e vídeo. Em relação às questões de reserva de recursos e garantia de serviços, o protocolo RTP não suporta tais funções relegando as mesmas para protocolos de mais baixo nível. Em conjunto com o RTP opera o protocolo RTP Control Protocol (RTCP) [11], usado para a recolha de informação estática das conexões media e controlo de geração de relatórios de *media*.

2.4 Resumo

As redes de telecomunicações encontram-se em crescimento exponencial, verificando-se por diversas vezes durante a sua evolução fases de revolução, sendo o aparecimento das redes IMS uma dessas fases. Através do IMS pretende-se alcançar níveis de serviço até à data inalcançáveis aumentando o portefólio de serviços prestados pelos operadores. Tirando proveito de um conjunto alargado de serviços com requisitos de qualidade e disponibilidade de serviço, o operador consegue apresentar um sistema verdadeiramente ubíquo atingindo os requisitos desejados pelos utilizadores. A possibilidade de desenvolvimento de uma solução deste tipo deve-se em grande parte aos esforços de entidades de normalização como o 3GPP, a ITU-T e a ETSI que através de diferentes normalizações apresentaram arquiteturas extensíveis e capazes de dar resposta aos requisitos do mercado.

Capítulo 3

Automatização de Testes

No capítulo 3 são discutidas as temáticas relacionadas com a automatização dos testes e a dificuldade de realização dos mesmos. Inicialmente é feita uma breve descrição dos diferentes tipos de testes a ser executados sobre uma determinada solução, secção 3.1 . De seguida são identificadas as problemáticas associadas com a automatização de testes SIP, secção 3.2. Por fim, são descritos os equipamentos SIP que serão usados para as validações, secção 3.3.

3.1 Taxonomia de Testes

Os testes são uma parte integrante de todos os projetos de desenvolvimento e/ou configuração, estando responsáveis por verificar/validar se o funcionamento do serviço/produto vai de encontro aos requisitos indicados pelo cliente. Embora no desenvolvimento de qualquer solução exista uma fase denominada por "fase de testes", existe a necessidade de executar testes em diferentes fases e sobre diferentes pretextos. Para cada uma das diferentes fases os testes têm características diferentes apresentado nomenclaturas distintas. De entre os diferentes tipos de testes destacam-se os seguintes.

3.1.1 Testes de Compatibilidade

Os testes de compatibilidade são testes não-funcionais, realizados com o intuito de validar o comportamento da solução num ambiente distinto do ambiente usado no decorrer do desenvolvimento. Validar soluções de *software* em ambientes distintos pode corresponder à verificação do

comportamento da aplicação desenvolvida sobre diferentes versões do mesmo sistema operativo. De entre os testes de compatibilidade, os testes realizados para validação de aplicações *web* sobre diferentes *browsers* são um dos exemplos mais reconhecíveis.

3.1.2 Testes de Sanidade

Os testes de sanidade correspondem a testes rápidos realizados com o intuito de validar se a solução se encontra a operar de forma expectável. Estes são constituídos por um número pequeno testes, tendo como principal intuito validar se a solução se encontra com maturidade suficiente para sofrer um conjunto de testes mais alargado. Caso seja determinado que a solução a validar cumpre com os requisitos apresentados na fase de sanidade, é realizado um conjunto de testes mais extenso e rigoroso.

3.1.3 Testes de Regressão

Os testes de regressão têm por objetivo encontrar diferenças no modo de funcionamento de um *software*/serviço comparativamente com uma versão anterior do mesmo. Por norma, os testes de regressão são aplicados para validar alterações de *software* ou *hardware* verificando se o comportamento expectável ainda se verifica. A importância dos testes de regressão destaca-se quando existem sinergias entre o equipamento a validar e a restante rede, sendo necessário garantir que os *inputs* e *outputs* esperados entre os diferentes equipamentos se mantêm idênticos às versões anteriores.

3.1.4 Testes de Aceitação

Os testes de aceitação são testes realizados pelo cliente após a validação por parte da empresa que desenvolveu o serviço. Normalmente os testes de aceitação envolvem todas as funcionalidades do sistema de forma a cobrir todos os casos de uso desenvolvidos. Habitualmente apresentam-se como testes complexos que têm por objetivo testar a solução num cenário o mais próximo possível do final, de forma a verificar como a solução na sua globalidade irá se comportar.

3.1.5 Testes de Desempenho

Os testes de desempenho têm por objetivo verificar o comportamento do sistema em situações de carga, validar a estabilidade, verificar a escalabilidade e analisar a tolerância a falhas. De entre as diferentes vertentes, os testes de carga são o grupo mais reconhecidos, tendo por objetivo colocar o sistema sobre *stress*, aumentando o número de utilizadores, operações e/ou ligações. Através dos testes de carga pretende-se validar requisitos não-funcionais da solução, verificando até que ponto a plataforma em situação de carga mantém os serviços com a qualidade expectável.

3.2 Automatização de Testes SIP

A qualidade de um projeto encontra-se diretamente relacionada com a capacidade dos testes definidos abrangerem a totalidade dos requisitos indicados pelo cliente da solução. Este é um princípio base que tem de ser levado em conta tanto durante o desenvolvimento como ao longo do tempo de vida de qualquer projeto. Contudo, a validação dos requisitos de forma manual apresenta-se como um processo moroso e sujeito a erros, sendo preferível executar os testes através de processos automáticos.

A utilização de testes automáticos permite aumentar a frequência com que os testes são executados, melhorar a utilização de recursos, diminuir o tempo despendido em validações e aumentar a confiabilidade na solução criada. Verificando-se um conjunto alargado de vantagens associadas à automatização de testes, percebe-se o porquê das empresas estarem a modificar os métodos de validação, tornando os testes cada vez mais automáticos. Embora as vantagens sejam diversas e o caminho a seguir passe pela automatização dos testes, a validação de testes de forma automática apresenta um conjunto vasto de questões que necessitam de ser endereçadas. Destacando-se as questões relacionadas com a definição dos testes, criação, execução e validação.

No caso prático do desenvolvimento e manutenção de soluções de redes IMS, os requisitos apresentam-se com frequência de elevada complexidade e em elevado número. Na maioria das ocasiões, os requisitos são de alto nível, não sendo possível de forma direta mapeá-los em testes capazes de serem realizados por ferramentas automáticas de validação. Relativamente ao número de requisitos, este é calculado tendo em conta os requisitos definidos pelo cliente multiplicado pelo número de diferentes acessos presentes na solução. Ou seja, os diferentes requisitos necessitam de ser validados sobre todos os diferentes acessos presentes na solução (ligações GPON,

ligações ADSL, ligações VPN, etc.), sendo pois, todos os requisitos validados em diferentes ocasiões sobre diferentes ambientes.

No que respeita a questões temporais, cada solução IMS é validada em ocasiões distintas com intuitos diversificados. Quando a solução é desenvolvida, a solução é alvo de testes de forma a validar se o comportamento apresentado é o comportamento expetável (testes funcionais). Antes de implementar a solução é realizado um subconjunto de testes de forma a verificar se a solução comporta-se em conformidade com o esperado (testes de sanidade). Seguidamente à validação do subconjunto de testes, é necessário validar a solução como um todo sobre um ambiente o mais similar ao ambiente final (testes de aceitação). Após a entrega da solução e no decorrer do tempo de vida do projeto é necessário repetir a validação dos testes sempre que existam alterações tanto a nível de *software* como de *hardware* no equipamento (testes de regressão).

Como se verifica, existe um conjunto extenso de situações onde é necessário realizar validações sobre a solução, sendo imperativo garantir não só que os testes são realizados de forma automática, mas também que a sua execução decorra no menor tempo possível.

A necessidade dos testes possuírem tempos de execução baixos justifica-se pela quantidade dos testes existentes. Uma bateria de testes pode com facilidade atingir trezentos testes por acesso. A título de exemplo, caso a solução possua três acessos distintos (acessos GPON, acesso ADSL e VPN dedicadas) são novecentos testes. Se cada teste demorar em média um minuto a preparar e executar são quinze horas de testes, verificando-se que mesmo em horário pós-laboral a execução dos testes apresenta uma janela temporal muito grande. Relativamente à execução dos testes, é necessário garantir que os comportamentos executados na validação (*inputs*, *outputs*) são similares aos que são verificados na rede onde o equipamento se encontra implementado. A necessidade de replicar comportamentos deve-se ao facto de se validar as soluções na rede de desenvolvimento, mas com os comportamentos existentes na rede do cliente.

Analisando todas as questões associadas com a automatização de testes, agrupou-se as mesmas em três temáticas distintas, criação de cenários, criação de testes e execução de testes.

A criação do cenário endereça todas as questões associadas com a preparação do equipamento a validar. Sobre a criação de cenários são abordadas todas as questões relacionadas com a obtenção das configurações, validação de informação de rede e validação da informação do cenário. A criação de testes acautela questões como a obtenção dos comportamentos do equipamento na rede (*input* e *outputs*), a geração de testes com base em comportamentos de rede e a inserção das validações nos testes. A execução dos testes responde às questões de obtenção dos *scripts* de testes, execução da bateria de testes, recolha de evidências e geração de relatórios.

Estes três tópicos são abordados em maior detalhe nas secções seguintes.

3.2.1 Criação de Cenários

A proposta de solução de automatização de testes IMS baseia-se na verificação de *outputs* tendo conhecimento dos *inputs* enviados. Neste tipo de soluções existe a necessidade de controlar os valores de entrada e de saída do equipamento a validar, sendo necessário controlar o ambiente de execução dos testes. A necessidade de controlar o ambiente de testes afasta a hipótese de validação das soluções na rede do cliente, tendo as validações de ocorrer na rede de desenvolvimento.

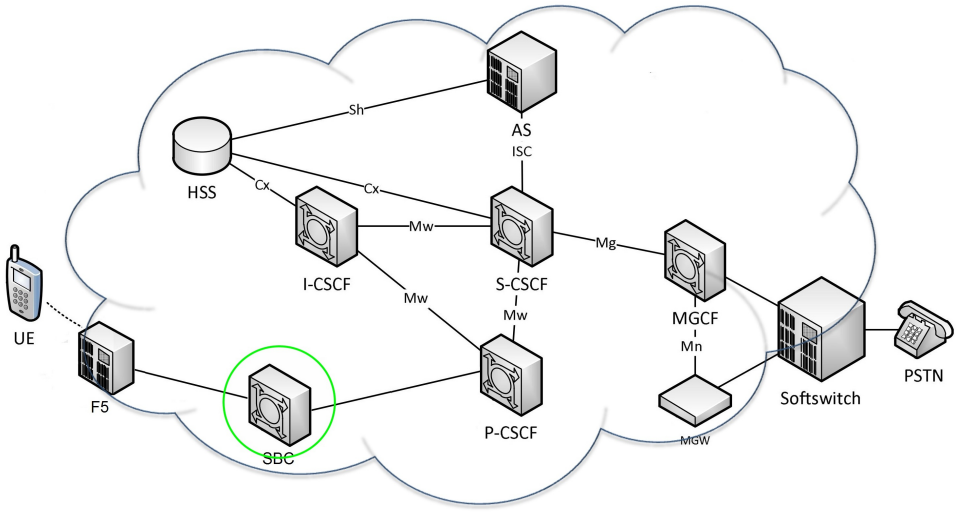


Figura 3.1: Arquitetura típica de rede IMS presente na rede do cliente

Um exemplo de uma arquitetura de rede de cliente assemelha-se à Figura 3.1, apresentando um conjunto de equipamentos distintos sobre os quais é necessário realizar validações. Por norma, este tipo de arquiteturas são constituídas por equipamentos de diferentes fornecedores, tendo equipas de suporte distintas para cada equipamento. Neste sentido, e como ilustrativo, caso se pretenda testar o SBC nas mesmas condições do que, o presente na Figura 3.1 é necessário controlar os *inputs* vindos do F5 e controlar os *outputs* disponibilizados ao P-CSCF. Ou seja, para além de controlar o equipamento a validar é necessário controlar os equipamentos circundantes, verificando-se dificuldades na realização de testes automáticos nestas condições. Contudo, para validar um equipamento na rede de desenvolvimento é necessário preparar a rede para a mesma operar com o equipamento.

O primeiro passo na preparação do cenário esta associado com a realização de configurações de rede em equipamentos com funções de *routing*. A configuração do equipamento sem a preparação prévia da rede torna o desenvolvimento da solução de automatização inviável. Todas as configurações de acessos, IPs e Virtual Local Area Networks (VLANs) que serão usadas no decorrer dos testes devem ser acauteladas, para que, quando se colocarem as configurações no equipamento este opere pelo menos ao nível de rede.

Após preparação da rede é necessário alterar as configurações de serviço. Embora se pretenda realizar testes com as configurações presentes na rede do cliente, com frequência é necessário ativar ou desativar configurações conforme os testes que se pretende realizar. Sendo necessário conhecer o funcionamento dos equipamentos, para alterar as configurações sem interferir no modo como os serviços são disponibilizados. Após o equipamento estar configurado é necessário abstrair-se do equipamento e analisar os comportamentos associados aos diferentes *inputs*. Para tal, é necessário conhecer os comportamentos espectáveis e não espectáveis para os diferentes *inputs*, de modo a poder realizar validações.

No ambiente de desenvolvimento existe por norma uma rede completa onde os equipamentos se encontram integrados. Contudo, no âmbito da validação de soluções, a utilização da rede na íntegra não se apresenta como uma solução vantajosa pois, é preferível isolar o equipamento e emular tráfego, aumentando assim o controlo dos fluxos de sinalização a validar. A não utilização de equipamentos com funções idênticas aos equipamentos presentes na rede do cliente justifica-se pela diversidade existente nas implementações dos protocolos entre diferentes fabricantes. Com frequência, em casos específicos, verifica-se equipamentos normalizados com funções similares a operar de forma diferente. As divergências existentes ao nível protocolar reforçam a necessidade de executar as validações em ambientes isolados, onde, existe controlo dos equipamentos que geram os *inputs* e *outputs*. Deste modo é necessário aplicar métodos que permitam gerar tráfego e injetar este no equipamento, replicando assim os comportamentos existentes na rede do cliente. Ou seja, para o caso específico do equipamento SBC, é necessário implementar um cenário semelhante ao presente na Figura 3.2, onde o equipamento é isolado e os *inputs* e *outputs* são enviados e recebidos por ferramentas de emulação de tráfego.

3.2.2 Criação de Testes

Os testes desenvolvidos para um sistema manual, necessitam de ser adaptados ou mesmo redesenhados de forma a operarem de modo automático. Para o ser humano torna-se simples seguir um conjunto de indicações e entender os requisitos e os comportamentos adjacentes. Contudo,

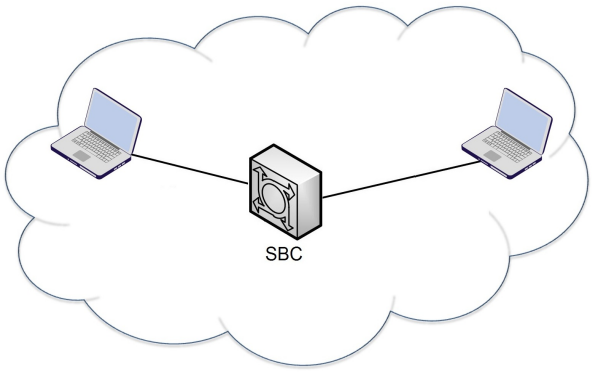


Figura 3.2: Modelo seguido na rede de desenvolvimento para validação do equipamento

quando os testes são automatizados necessitam de operar a um nível mais baixo, executando validações voltadas para os valores obtidos sabendo os comportamentos iniciais. Não possuindo as ferramentas de validação a capacidade de análise e interligação da informação como o ser humano, torna-se necessário desenvolver um caderno de testes mais rudimentar e direcionado para execução de testes funcionais.

A geração de testes deve entrar em conta com o comportamento dos equipamentos circundantes ao equipamento a validar pois, as configurações do equipamento são elaboradas tendo em conta as interferências da rede. No decorrer dos testes pretende-se validar que um determinado requisito gera um comportamento com valores específicos. Contudo, o mesmo requisito em equipamentos de fabricantes distintos pode apresentar comportamentos distintos, comprometendo assim os testes.

Imagine-se que se pretendia validar o S-CSCF2 presente na Figura 3.3. Inicialmente era necessário validar o fluxo de estabelecimento de sessões, analisando as mensagens SIP enviadas entre o I-CSCF2 e o S-CSCF2, sendo posteriormente, analisadas as mensagens geradas entre o S-CSCF2 e o P-CSCF2.

Ao nível dos fluxos, quando as mensagens chegam ao S-CSCF2, este deve possuir a capacidade de encaminhar as mensagens para o P-CSCF2, analisando a informação que se encontra na mensagem atual recebida.

O fluxo típico de iniciação de uma sessão é similar ao presente na Figura 3.4, havendo um conjunto de transações desde o envio da mensagem pelo UE1 até à chegada ao UE2. Ao longo do fluxo verifica-se que a mensagem enviada pelo UE1 sofre alterações, sendo a mensagem recebida no UE2 circunstancialmente diferente da enviada. Relativamente às mensagens trocadas com o

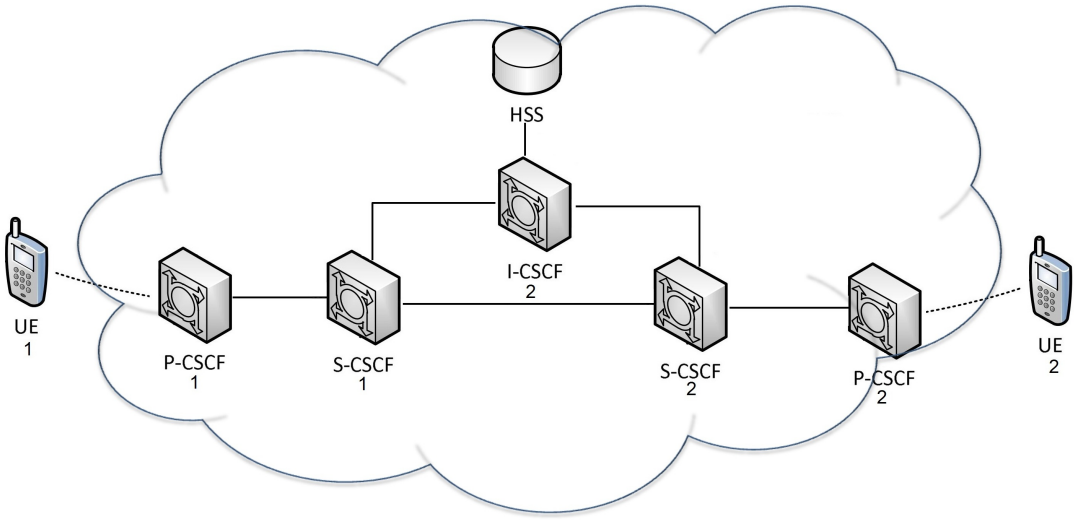


Figura 3.3: Estabelecimento de sessão entre dois terminais na mesma *home network*

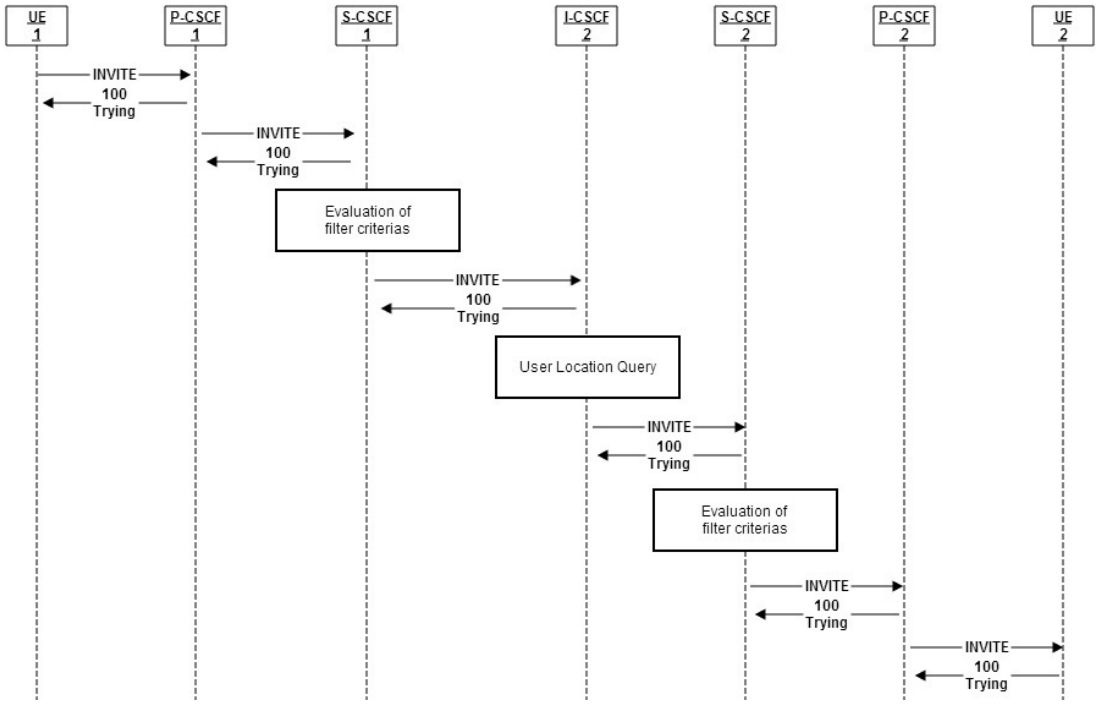


Figura 3.4: Fluxo inicial tido no estabelecimento de uma sessão SIP

S-CSCF2, estas apresentam-se similares à Figura 3.5 e Figura 3.6. Sendo facilmente percetível que a mensagem enviada pelo I-CSCF e a recebida pelo P-CSCF apresentam-se com muitas diferenças.

A análise dos comportamentos na rede e a captura de fluxos associados ao equipamento a validar (no caso do exemplo o S-CSCF2) apresentam-se como tarefas relevantes para a validação da solução pois, é sobre os fluxos que devem ser gerados os *scripts* para as ferramentas de emulação. Os *scripts* devem representar claramente os comportamentos da rede, replicando os campos presentes na sua totalidade. Embora a criação dos fluxos aparente ser simples devido o formato textual das mensagens, a criação destes encontra-se sujeita a erros pelo tamanho extenso das mensagens verificando-se a necessidade de automatizar o processo de geração dos *scripts*. Deste modo, diminui-se os possíveis erros de criação do fluxo como também o tempo despendido na geração dos *scripts*.

```
INVITE sip:user2_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home1;netbranch=z9hG4bK871y12.1,
 SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1,
 SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK431h23.1,
 SIP/2.0/UDP (5555::aaa:bbb:ccc:ddc1):1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 67
Route: <sip:scscf2.home1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net;lr>
P-Asserted-Identity: "John Doe" <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6021rT5tAFrbHLso=023551024"; orig-ioi=home1.net
Privacy: none
From: <sip:user1_public1@home1.net>;tag=171828
To: <tel:+1-212-555-2222>
Call-ID: cb03a0s09a2sdfglkj490333
Cseq: 127 INVITE
Require: precondition
Supported: 100rel
Contact: <sip:[5555::aaa:bbb:ccc:ddc1]:1357;comp=sigcomp>
Allow: INVITE, ACK, CANCEL, BYE, PRACK, UPDATE, REFER, MESSAGE
Content-Type: application/sdp
Content-Length: 328

[SDP]
```

Figura 3.5: Mensagem entre o I-CSCF2 e o S-CSCF2 (input do S-CSCF2)

Juntamente com a geração dos *scripts* a executar na ferramenta de emulação existe a necessidade de inserir métodos de validação dos comportamentos. Um caso prático do tipo de validações que necessitam de ser realizadas encontra-se demonstrado na Figura 3.7. Esta análise parte do pressuposto que se conhece sempre o que seria expectável receber, sendo necessário para cada teste indicar e validar quais seriam os valores expectáveis. Desta forma é necessário garantir o comportamento dos *inputs*, obter os *outputs* e validar os mesmos. No caso prático presente na Figura 3.7 verifica-se que o valor do *user part* do From esperado era user1_public. Contudo, o valor recebido foi bob_public1 tendo o teste de ser marcado como falhado.

```
INVITE sip:(5555::eeelfhaaa:bbb]:8805;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf2.home1.net;branch-=z9hG4bK764z87.1,
 SIP/2.0/UDP icscf2_s.home1.net;branch=z9hG4bK871y12.1,
 SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1,
 SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK431h23.1,
 SIP/2.0/UDP (5555::aaa:bbb:ccc:ddd]:1357:comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 66
Route: <sip:pcscf2.home1.net;lr>
Record-Route: <sip:scscf2.home1.net;lr>, <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net;lr>
P-Asserted-Identity: "John Doe" <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6021rT5tAFrbHLso=023551024"
Privacy: none
From: <sip:use1_public1@home1.net>;tag=171828
To: <tel:+1-212-555-2222>
Call-1D: cb03a0s09a2sdfg1kj490333
Cseq: 127 INVITE
Require: precondition
Supported: 100rel
Contact: <sip:(5555::aaa:bbb:ccc:ddd):1357:comp=sigcomp>
Allow: INVITE, ACK, CANCEL, BYE, PRACK, UPDATE, REFER, MESSAGE
P-Called-Party-1D: <sip:user2_public1@home1.net>
Content-Type: application/sdp
Content-Length: 340

[SDP]
```

Figura 3.6: Mensagem entre o S-CSCF2 e o P-CSCF2 (output do S-CSCF2)

```
From received: sip:bob_public1@home1.net;tag=12345
User part received: bob_public1
User part expected: user1_public1
Test Result: NOK
```

Figura 3.7: Validação do *user part* do cabeçalho From

3.2.3 Execução de Testes

A execução de testes de forma manual como indicado anteriormente é um processo bastante moroso. Inicialmente, para validar um teste é necessário realizar as configurações dos terminais, verificar acessos ao equipamento e preparar máquinas de captura. Após ter todas as condições para a realização e captura do teste, decorre a execução do mesmo. Por fim é necessário fazer a filtragem das evidências, analisar cabeçalhos específicos das mensagens trocadas, reportar sucesso/insucesso do teste e armazenar as evidências para futura análise.

Todos os processos envolventes na execução de um teste fazem com que o tempo médio despendido para a realização de cada teste seja entre 40 a 60 minutos. Com um tempo médio tao elevado para a execução de um teste verifica-se desde logo, a necessidade de automatizar o processo de execução. Contudo, o processo de execução de testes apresenta dificuldades intrínsecas, sendo necessário desenvolver mecanismos capazes de controlar e indicar que testes realizar, como os realizar e como sincronizar todo o processo.

O equipamento que se pretenda validar pode operar de diferentes formas em diferentes momentos, tendo a solução de ser capaz de acompanhar as necessidades dos testes. Ou seja, o

equipamento pode-se apresentar como um elemento de passagem, existindo para um dado *input* um *output* específico, pode operar como equipamento terminal recebendo somente *inputs* ou, pode enviar *outputs* sem receção previa de *inputs*. Todos estes casos de execução necessitam de ser acautelados, sendo os mais preponderantes e de difícil análise os casos de receção de *inputs* e envio de *outputs* devido às questões de sincronismo subjacentes. Para além das questões de sincronismo, a ferramenta de execução tem de acautelar a captura de evidências sobre os comportamentos verificados em cada teste e gerar relatórios sucintos sobre a execução de cada teste.

3.3 Equipamentos sob Processo de Automatização

A solução a desenhar tem como intuito ser capaz de operar com um conjunto diversificado de equipamentos das redes IMS. Contudo, no decorrer do trabalho, as verificações e validações de testes vão ocorrer sobre os SBCs da Oracle podendo estes ser auxiliados pelo *Load Balancer* da F5. A necessidade de validação destes elementos despoletou a proposta de solução desenvolvida, estando contudo, a proposta pensada de forma a abranger um maior espectro de equipamentos.

3.3.1 Session Border Controller

O Session Border Controller (SBC) [59] é um elemento das redes IMS com funções de controlo de sinalização e de *media*, sendo usado por norma para a separação entre a rede de acesso e a rede *core*. O papel do equipamento na rede é preponderante devido ao controlo que aplica sobre todo tipo de sessões (tempo real, interativas, de voz, de vídeo), endereçando questões críticas para os operadores nas áreas de segurança, interoperabilidade e qualidade de serviço.

Devido à sensibilidade dos dados passados na rede, é essencial que todos os elementos da rede tenham presentes preocupações a nível da segurança, não sendo os SBCs exceção. O equipamento possui funções de controlo de acessos, policiamento de fluxos, gestão de tráfego, proteção de ataques de Denial of Service (DoS), funções de autenticação de utilizadores e funções de encriptação de ligações. Os SBCs são equipamentos tipicamente de fronteira de rede, endereçando muitas questões de interoperabilidade entre domínios. As questões de interoperabilidade são relevantes pois, nem todos os domínios usam o mesmo protocolo de sinalização, sendo necessário fazer a interligação de forma a permitir as comunicações entre as diferentes redes. Relativamente aos requisitos de qualidade de serviço, estando os SBCs associados à gestão de *media*, estes necessitam de ter preocupações associadas com o controlo de tráfego, atribuição de prioridades e

diferenciação de tráfego.

Ao nível das configurações, os SBCs possuem as suas funções separadas em seis ramos distintos como apresentado na Figura 3.8.

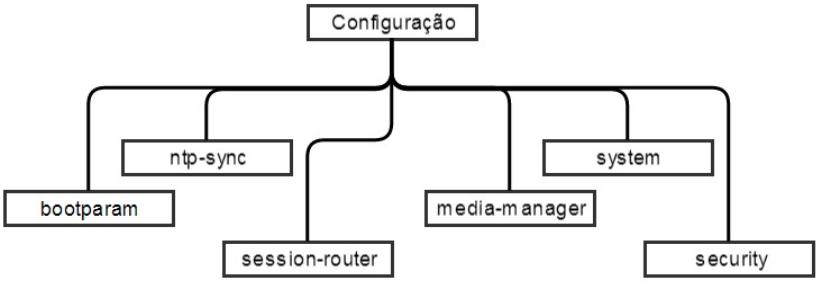


Figura 3.8: Ramos de configuração do SBC

O bootparam possui as configurações de *boot* do equipamento, tendo informação sobre o *firmware*, endereço de O&M, entre outros. O ramo ntp-sync contém as configurações dos servidores de Network Time Protocol (NTP). O security tem as configurações associadas à segurança de rede e dos clientes, englobando questões de autenticação de utilizadores, encriptação de mensagens e controlo de fluxos. O Session-router possui todas as configurações de sinalização, desde configurações das interfaces de sinalização, às configurações de políticas de encaminhamento. O media-manager tem todas as configurações relacionadas com o *media*, desde configurações dos realms, até às configurações de DNS. O system possui configurações de rede como as interfaces de rede ao nível IP, configurações de monitorização do estado do equipamento, entre outras.

Em termos de encadeamento das configurações, estas são integradas de forma similar à presente na Figura 3.9. As physical interfaces correspondem às interfaces físicas do equipamento podendo ser de dois tipos distintos, media interfaces ou management interfaces. As management interfaces são usadas para funções de O&M, como acesso remoto ao equipamento, transferência de ficheiros e obtenção de informação de monitoria. As media interfaces são usadas para passagem de tráfego de sinalização e *media*, sendo sobre estas definidas as network interfaces. Ao nível das network interfaces, são configuradas a informação de nível 3, indicando configurações de VLANs e de IPs.

Ao nível do *media*, para se conseguir estabelecer sessões é sempre necessário configurar no mínimo os realms e as sterring-pools. O realm define-se como uma entidade lógica constituídas por uma ou mais redes, sendo responsável pela gestão de sessões em tempo real. Tendo em conta as configurações dos SBC, os realms apresentam-se como um elemento central agregador

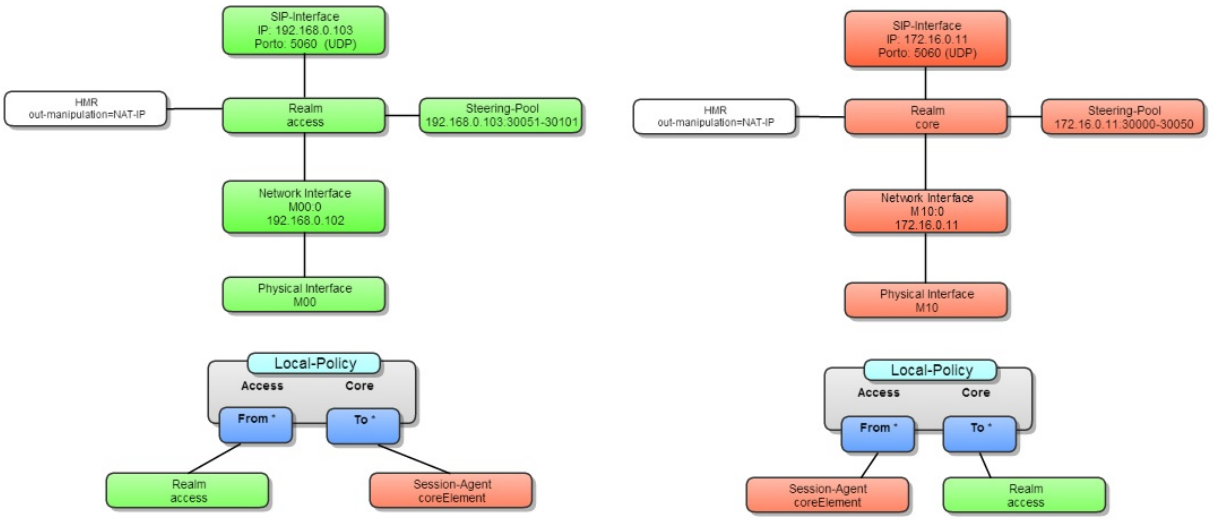


Figura 3.9: Configurações essenciais no SBC

das configurações de sinalização e *media*. No que diz respeito ao *media*, os realms operam em conjunto steering-pools para garantir o endereçamento IP e a definição de portas para as ligações operarem com sucesso. As steering-pools encontram-se associadas a um realm e a um IP, possuindo um conjunto alargado de portas para o estabelecimento de sessões RTP.

Ao nível da sinalização, por norma é necessário realizar configurações de sip-interfaces, session-agents, local-policies e sip-manipulations (HMRs). As sip-interfaces apresentam-se como configurações obrigatórias no SBC, sendo usadas para identificação do IPs:Portos através dos quais se pretende obter informação de sinalização. Os session-agents apresentam-se como configurações opcionais usadas para a identificação de um próximo nó para atingir a rede de *core* ou acesso. Relativamente ao encaminhamento entre a rede de acesso e *core* no SBC, este pode ser realizado através do uso de local-policies que vão permitir a realização do encaminhamento das mensagens ao nível da sinalização. Para a manipulação das mensagens são usadas as sip-manipulations permitindo a realização de modificações nos fluxos, cabeçalhos ou parâmetros das mensagens.

Um cenário simples da implementação de configurações do SBC apresenta-se similar ao ilustrado na Figura 3.10. Neste figura, encontra-se descrito um cenário *peering* onde se encontra configurado um acesso e um *core*, sendo o SBC o elemento de interligação.

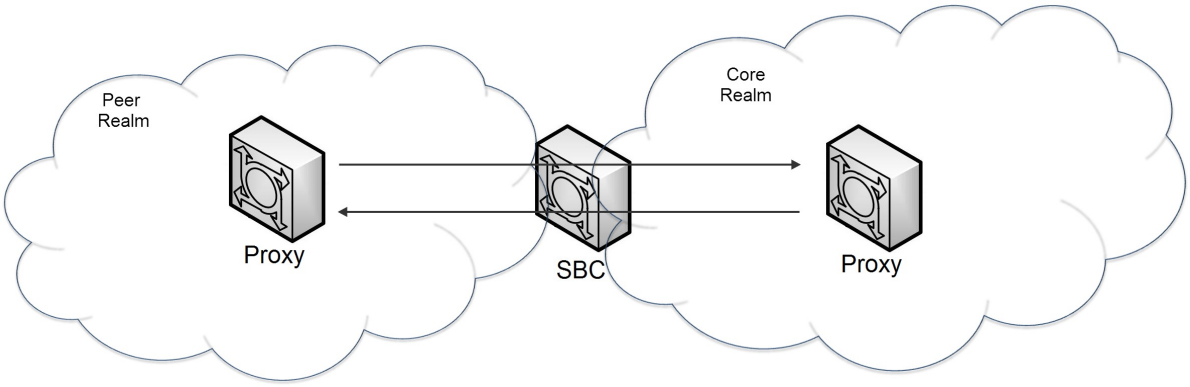


Figura 3.10: Diagrama de arquitetura *peer-to-peer*

3.3.2 F5 BIG-IP

O BIG-IP apresenta um conjunto de aplicações para melhorar a disponibilidade, desempenho e segurança dos sistemas de rede [60]. Das funções disponibilizadas pelo BIG-IP destacam-se as funcionalidades de encaminhamento de diferentes tipos de tráfego para servidores distintos, fazendo a distribuição de tráfego sobre diferentes algoritmos. O balanceamento dos serviços de modo diferenciado é realizado através do módulo Local Traffic Manager (LTM), que é responsável pelas funções associadas à gestão do tráfego que entra ou sai da rede LAN. Este módulo permite fazer a interceção e redireccionamento de tráfego, tendo funções de manipulação de cabeçalhos, gestão de certificados Secure Sockets Layer (SSL) e filtragem de pacotes [61]. Ao nível de configurações, o módulo LTM tem como elementos básicos os virtual servers, as load balancing pools e os profiles. As configurações do módulo LTM só são possíveis depois da realização das configurações do nível de rede, sendo necessário configurar VLANs, routes e self IPs.

Ao nível de configurações de rede, as VLANs são as configurações iniciais realizadas no equipamento, sendo necessário mapear as VLANs associadas a cada interface. Após realização das configurações de VLANs é necessário restringir ou agregar domínios de *routing*, definindo as configurações dos route domains. Para cada VLAN criada é recomendável associar um IP ao BIG-IP, sendo o IP especificado através das configurações de self IPs. Ao nível das routes, são configuradas rotas de encaminhamento, podendo as mesmas ser definidas para um IP ou para uma *gateway*.

Dentro do módulo LTM, os virtual servers apresentam-se como elementos de elevada relevân-

cia para a distribuição de tráfego, manipulação de cabeçalhos e controlo de ligações SSL. No que diz respeito às pools, estas apresentam-se como elementos agregadores dos dispositivos sobre os quais se pretende fazer a distribuição de tráfego. A distribuição de tráfego sobre os pool members pode ser realizada seguindo métodos distintos, como balanceamento por prioridades, round robin, least sessions, entre outros. Os pools members são os elementos sobre os quais se pretende balancear o tráfego. Estes, por norma, são definidos nas configurações do LTM como nodes indicando o IP das entidades. Nas configurações de profiles é possível fazer a gestão do serviço, tendo para cada serviço um conjunto melhorado de funcionalidades de gestão. No que diz respeito à monitorização, todas as configurações de LTM podem ser verificadas através da análise de monitors. Os monitors podem ser adicionados aos elementos como nodes, pools e virtual servers sendo usados para identificar o estado do equipamento em tempo real.

Em termos arquiteturais, no âmbito do trabalho desenvolvido, o BIG-IP apresenta-se como um sistema complementar ao SBC, sendo usado para fazer o balanceamento do tráfego entre dois ou mais SBCs, como demonstrado na Figura 3.11.

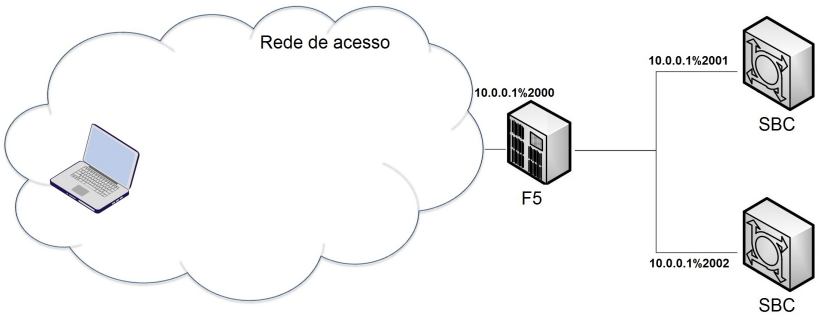


Figura 3.11: Diagrama de arquitetura F5 BIG-IP

3.4 Resumo

Embora a automatização dos testes seja o caminho a seguir de modo a tornar mais ágeis os processos morosos, verifica-se a existência de um conjunto diverso de questões que necessitam de ser consideradas. Contudo, após ter as diversas questões endereçadas e a solução de testes desenvolvida torna-se rápido o processo de validação, conseguindo dar uma resposta célere ao cliente.

No que diz respeito aos equipamentos, a proposta é pensada de forma a operar sobre um

conjunto diverso de equipamentos. Contudo, no âmbito deste trabalho, a validação da proposta será realizada sobre os equipamentos Session Border Controllers disponibilizados para testes pela PT Inovação.

Capítulo 4

Replicação de Cenários e Automatização de Testes: Solução Proposta

Neste capítulo apresenta-se a proposta de solução, indicando a arquitetura planeada, as aplicações desenvolvidas e os procedimentos desenhados. Inicialmente descreve-se a arquitetura da solução na secção 4.1. De seguida é apresentada a ferramenta de emulação a usar (secção 4.2). Na secção 4.3 descreve-se a preparação do cenário e as aplicações adjacentes. Os processos de criação de testes são apresentados na secção 4.4. Por fim, apresenta-se o processo de execução de testes na secção 4.5.

4.1 Arquitetura global do Sistema de testes

A proposta de solução apresentada para a automatização dos testes em redes IMS envolve questões diversas, que vão desde a preparação dos equipamentos até à execução dos testes, verificando-se um conjunto alargado de temáticas subjacentes à proposta. Tendo a automatização de testes um âmbito de atuação tão alargado, definiu-se uma arquitetura modular idêntica à apresentada na Figura 4.1, separando as temáticas distintas por módulos diferentes cooperantes entre si. O módulo de criação de testes trata da geração de testes a partir de informação capturada na rede do cliente. A criação de cenários endereça as questões relacionadas com a preparação do equipamento da rede de desenvolvimento com as configurações da rede do cliente. O módulo de execução de testes endereça questões associadas com a execução das baterias de testes e recolha de evidências. Cada um dos módulos possui um conjunto de aplicações que foram desenvolvidas

de raiz ou modificadas para funcionarem de acordo com o pretendido, operando sobre diferentes linguagens em diferentes ambientes.

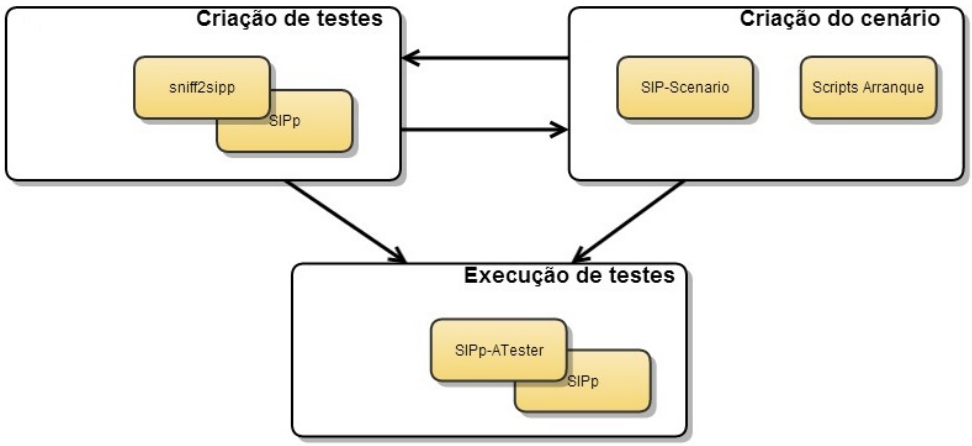


Figura 4.1: Diagrama de componentes da solução

Em conjunto com as aplicações desenvolvidas e modificadas foi ainda configurada uma ferramenta de emulação de tráfego SIP, de forma a atuar como base da execução e geração de testes. As iterações entre as diferentes aplicações e a ferramenta de geração de tráfego são definidas nos procedimentos que acompanham cada um dos módulos, explicando claramente quais os passos a seguir e que aplicações usar.

4.2 Emulação de tráfego SIP

A utilização de ferramentas de emulação de tráfego é preponderante para o sucesso da solução proposta, não fazendo parte do âmbito do trabalho desenvolver um processo de emulação de raiz. Através do uso de ferramentas já desenvolvidas e com provas dadas, consegue-se acelerar o processo de criação da solução, tendo por base uma ferramenta robusta. De entre as ferramentas analisadas optou-se pelo uso da ferramenta de emulação SIPp devido à sua aceitação por entidades relevantes na área do IMS como a Fraunhofer [62], pela possibilidade desta operar com equipamentos físico e por a ferramenta permitir realizar validações sobre os fluxos SIP.

4.2.1 SIPp

O SIPp é uma ferramenta *open source* de geração de tráfego SIP [63] caracterizada por ser de fácil uso para a emulação de cenários SIP através de documentos XML. Ao nível protocolar, o SIPp possui suporte para IPv4, IPv6, TLS, SCTP, entre outros protocolos, permitindo desenvolver cenários similares aos cenários reais. Para além da maleabilidade dos cenários SIP sobre diferentes protocolos, a plataforma permite a integração de equipamentos físicos nos fluxos dos testes, sendo para a plataforma indiferente o comportamento que os equipamentos apresentam (Proxy, B2BUA, Registrar, entre outros). Com o uso do SIPp pretende-se emular tráfego de uma entidade UAC para o equipamento a validar e analisar o *output* enviado para o UAS. Na Figura 4.2 mostra-se um exemplo da interação dos elementos SIPp com o equipamento SBC. Os elementos SIPp apresentados como SIPp UAC e SIPp UAS são na realidade dois documentos XML que serão executados pela ferramenta SIPp, sendo o formato dos ficheiros XML de simples interpretação, indicando quais as mensagens SIP a ser enviadas e quais as mensagens que são expectáveis receber.

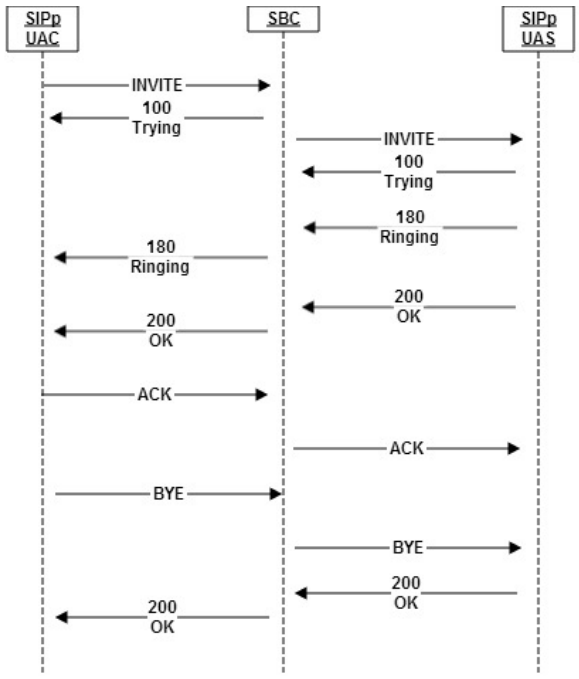


Figura 4.2: Fluxos no cenário de chamada por defeito do SIPp

Ao nível funcional, a informação é passada através de *tags*, podendo ser de dois tipo distintos: *send*, para o envio de pedidos/respostas; ou do tipo *recv* para a indicação do pedido/resposta a

receber. Para cada mensagem SIP enviada existem sempre as duas *tags* presentes em *scripts* distintos. Um caso prático corresponde ao INVITE enviado entre o SIPp UAC e o SIPp UAS presente na Figura 4.2. No *script* do UAC o INVITE é mapeado numa mensagem a enviar, sendo associado a uma *tag send*, enquanto que, no *script* UAS o INVITE é associado a uma *tag recv*. Com o mapeamento entre a mensagem enviada e a que se espera receber torna-se possível validar desde logo o fluxo, uma vez que, como apresentado na Figura 4.3, para cada *send* deve existir uma *tag recv*.

```
<send>                          | <recv request ="INVITE" >
INVITE …                        |
</send>                         | </recv>
```

Figura 4.3: Comparação do conteúdo da *tag send* com o presente na *tag recv*

No conteúdo das mensagens SIPp podemos encontrar informação estática do teste SIP e informação dinâmica. A informação estática corresponde a informação SIP que não necessita, ou deve ser alterada. A informação dinâmica corresponde a variáveis incrementais (Call-ID, tag, Branch) ou a variáveis identificativas dos utilizadores intervenientes na sessão. Em termos funcionais, a implementação de informação dinâmica é concretizada através do uso de variáveis SIPp, podendo ser de dois tipos: variáveis internas, como por exemplo as variáveis [service], [remote_ip], [call_number]; ou variáveis auxiliares [field*n*] definidas através de ficheiros CSV associados ao teste. Aquando da iniciação dos *scripts* SIPp indica-se qual o ficheiro que contém os *fields*, sendo os valores lidos do ficheiro de forma sequencial. A utilização de ambas as variáveis é em tudo similar como é verificado na Figura 4.4

```
INVITE sip:[field0]@[remote_ip] SIP/2.0
```

Figura 4.4: Método INVITE presente no *script* SIPp

Além de permitir a emulação de cenários, o SIPp permite também validar comportamentos através do uso de expressões regulares, podendo assim, validar cabeçalhos específicos nas diferentes mensagens. Em termos funcionais, as expressões regulares podem ser aplicadas tanto em *scripts* de UAC como em *scripts* de UAS, sendo em ambos os casos aplicadas sobre *tags ereg*. Um exemplo de uma expressão regular aplicada a um INVITE recebido é apresentado na Figura 4.5. Inicialmente é identificada a expressão a validar no parâmetro *regexp*. De seguida indica-se qual o local da mensagem onde se vai validar a expressão, podendo a validação ser na

totalidade da mensagem ou sobre algum cabeçalho específico. Caso seja um cabeçalho específico indica-se o mesmo no *header*. Por fim indica-se a variável onde armazenar o valor associado à expressão regular através do parâmetro *assing_to*.

```
<recv request="INVITE" crlf="true" >
 <action>
  <ereg regexp=".*" search_in="hdr" header="To" assign_to="1" />
  <log message="To is [$1]"/>
 </action>
</recv>
```

Figura 4.5: Exemplo de configuração de expressão regular

Para dar início aos *scripts*, recorre-se ao uso de comandos similares aos presentes na Figura 4.6. Inicialmente define-se o SIPp a executar (SBC_11_001-01.xml), de seguida o ficheiro com a informação auxiliar dos *fields* (SBC_11_001-01), o IP local e a porta de serviço (192.168.0.102, 5060), o número de execuções do teste (1), o IP e porta da entidade remota para onde a mensagem é encaminhada (192.168.0.104:5060) e por último, indica-se o tempo máximo que a execução pode possuir (UAC 5000 milissegundos , UAS 5 segundos).

```
//UAC
./sipp -sf SBC_11_001-01.xml -inf SBC_11_001-01 -i 192.168.0.104 -p 5060 -m 1
192.168.0.102:5060 -recv_timeout 5000

//UAS
./sipp -sf SBC_11_001-01.xml -inf SBC_11_001-01  -i 192.168.0.102  -p 5060 -m 1
192.168.0.104:5060 -timeout 5
```

Figura 4.6: Comandos de arranque dos scripts SIPp

No decorrer da execução dos testes é disponibilizada uma interface dinâmica similar à presente na Figura 4.7, dando informação do estado atual do teste, tempos de execução, número de chamadas com sucesso e insucesso, entre outros valores. A interface apresentada é constituída por um conjunto vasto de ecrãs, apresentando diferentes estatísticas sobre os testes.

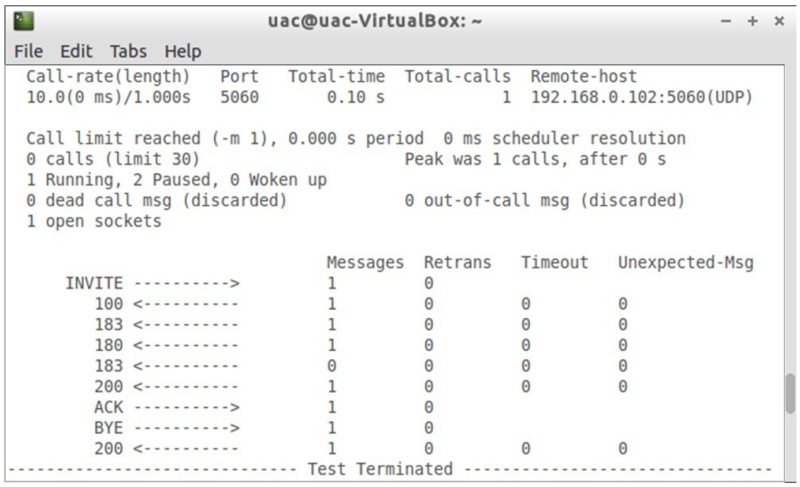


Figura 4.7: Shell de execução do UAC SIPp

4.3 Preparação do cenário de teste

A configuração do cenário engloba três fases distintas para a preparação do equipamento:

- Validação das configurações de encaminhamento
- Verificação das configurações dos equipamentos
- Alteração das configurações

O conjunto das três fases indicadas encontram-se representadas no diagrama da Figura 4.8 onde, como se pode observar, indica-se os diferentes passos a realizar para conseguir preparar o equipamento para operar.

As validações das configurações de encaminhamento consistem em testar se os acessos sobre os quais se pretende fazer testes já se encontram presentes na rede de desenvolvimento. Nestas validações, é necessário verificar a presença das configurações de VLANs e IPs nos diferentes equipamentos de encaminhamento. A título de exemplo, tendo um equipamento SBC com as configurações indicadas na Figura 4.9, tem de validar-se a presença das configurações de VLANs tanto de acesso como de *core*, verificar se o router faz o encaminhamento dos pacotes para os IPs indicados na configuração, validar se o tráfego não é bloqueado em nenhuma firewall e verificar se os switches encontram-se a etiquetar adequadamente os pacotes nas várias VLANs.

Uma vez validadas as configurações ao nível de rede, é necessário verificar e alterar as configurações dos elementos a validar e dos elementos auxiliares à validação. No caso específico do

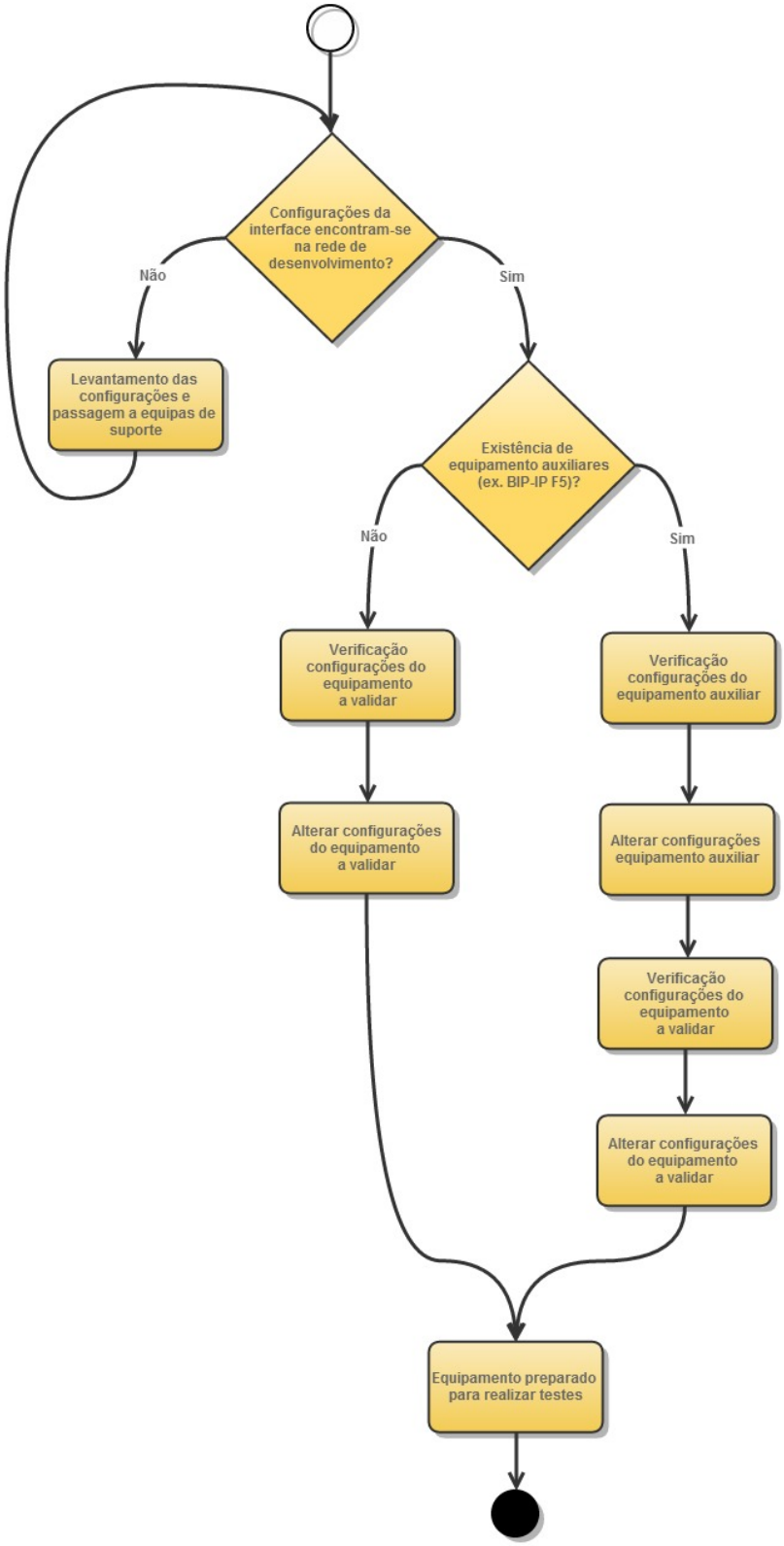


Figura 4.8: Procedimento para a criação de cenários para teste

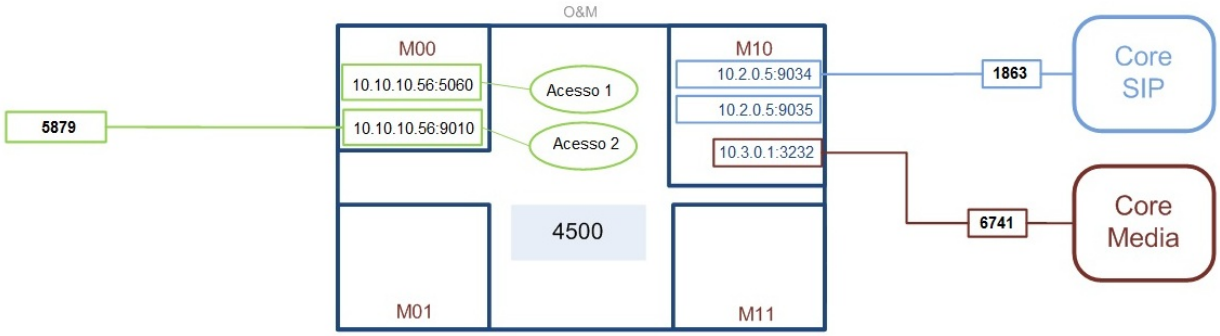


Figura 4.9: Arquitetura de rede de equipamento SBC

SBC, os testes de balanceamento são possíveis através do uso do F5, sendo necessário preparar e configurar ambos os equipamentos para a realização das validações.

Antes de realizar as alterações nas configurações dos equipamentos, é necessário proceder em primeiro lugar a um levantamento das configurações que necessitam ser alteradas, sendo a tarefa realizada na fase de verificação das configurações. No decorrer da verificação são analisadas as configurações que se sabe à partida que pode ser críticas para o sucesso da realização dos testes. Neste tipo de verificação valida-se o estado de configurações, identificando-se configurações a atualizar, inserir ou eliminar. Este tipo de verificações tem por base o conhecimento obtido pela experiência de validações manuais, sendo um conhecimento intrínseco de quem realiza testes e manipulações de configurações. De forma a tornar as verificações num processo acessível a um maior número de pessoas, desenvolveu-se a aplicação SIP-Scenario com o intuito de tornar o processo de levantamento das configurações mais transparente.

No que respeita às alterações de configurações, estas são realizadas através de *scripts* associados a clientes Secure Shell (SSH). No caso da solução proposta o cliente SSH escolhido é o extraPutty [64], essenvialmente pela sua capacidade de operar com *scripts* Lua [65]. A estes *scripts* dá-se o nome de *scripts* de arranque, sendo a preparação do equipamento para a execução de testes realizada através dos mesmos.

4.3.1 SIP-Scenario

O SIP-Scenario apresenta-se como uma aplicação capaz de fazer o levantamento das configurações a alterar de forma automática. Ao nível dos equipamentos, a solução encontra-se preparada para dar suporte aos equipamentos BIG-IP e SBC, permitindo contudo o futuro alargamento do tipo de equipamentos de suporte, dada a estrutura modular.

A lógica de funcionamento da ferramenta SIP-Scenario é demonstrada na Figura 4.10 que apresenta o diagrama de sequência associado à verificação das configurações dos diferentes equipamentos.

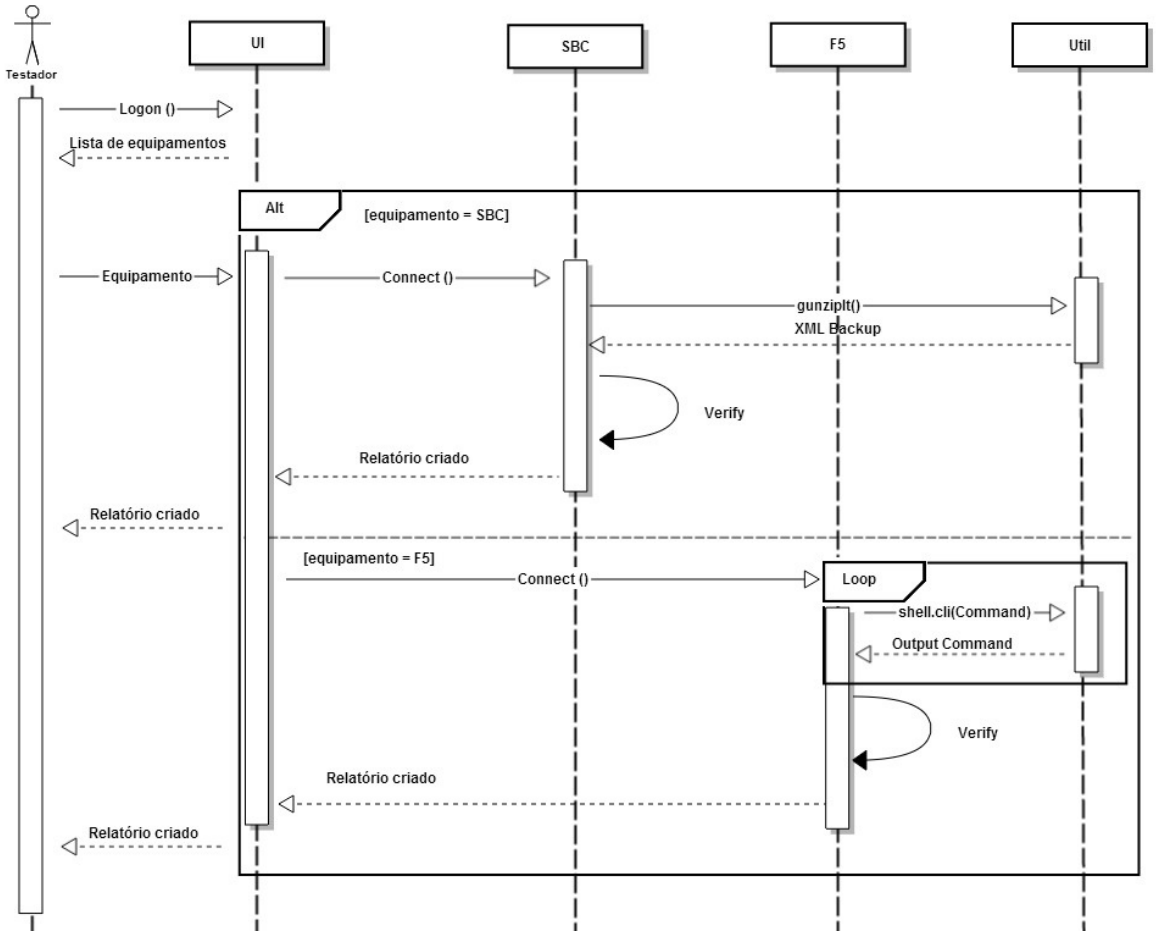


Figura 4.10: Diagrama de sequência da aplicação SIP-Scenario

Inicialmente, o utilizador executa a aplicação indicando informação do equipamento a validar. Com a informação em memória a aplicação analisa o *backup* passado, caso se esteja a validar o SBC, ou executa um conjunto de comandos para validação do BIG-IP. No decorrer da análise são verificadas as configurações na totalidade, analisando as configurações de rede como VLANs e IPs até às configurações especificas do equipamento como virtual servers e sip-interfaces. Através da análise realizada, a aplicação gera um relatório indicando as configurações que podem aparentar-se como críticas para a colocação do equipamento na rede de desenvolvimento.

Para o desenvolvimento desta aplicação optou-se pela linguagem de programação Java, pela independência do ambiente de execução e também pelo conjunto alargado de bibliotecas de suporte que a linguagem possui. Relativamente à aplicação desenvolvida, esta possui funções de execução de comandos e funções de análise de documentos XML, sendo usadas para a validação das configurações do F5 e do SBC. Na validação dos cenários dos SBCs é possível analisar os *backups* devido ao formato XML que os ficheiros apresentam, utilizando para tal bibliotecas Document Object Model (DOM).

Em termos funcionais, a obtenção de informação do documento XML é realizada seguindo o código presente na Figura 4.11. Inicialmente define-se a expressão a pesquisar no documento. Com a expressão definida são pesquisados todos os elementos do documento que possuem essa expressão, sendo retornada uma lista de elementos. Tendo a lista de elementos definida, analisam-se os parâmetros de cada um dos elementos, recolhendo a informação que se pretende verificar.

```
expression = "expression";
NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(xmlDocument, XPathConstants.NODESET);
for (int i = 0; i < nodeList.getLength(); i++) {
 String value = nodeList.item(i).getParentNode().getAttributes().getNamedItem("value").getNodeValue();

}
```

Figura 4.11: Processo de filtragem de documentos XML

No caso do elemento BIG-IP, não existe a possibilidade de analisar *backups*, o que leva à necessidade de executar um conjunto de comandos diretamente sobre o equipamento. Relativamente aos comandos executados, estes têm como objetivo validar as instancias do F5 como os virtual servers, pools, nodes, VLANs e self IPs. Os comandos executados são os presentes na Figura 4.12 sendo estes executados sobre um cliente SSH implementado no SIP-Scenario. Após obtenção dos valores é realizada a análise dos resultados obtidos, apresentando no final só a informação com valor para o utilizador.

```
bigpipe vlan all show
bigpipe route domain all vlan show
bigpipe self all show
bigpipe route all show
bigpipe node all monitor show
bigpipe pool all members all monitor show
bigpipe virtual  all show
```

Figura 4.12: Comandos executados para validação das configurações do BIG-IP

4.3.2 *Scripts* de Arranque

Os *scripts* de arranque são elaborados na linguagem Lua e executadas com o extraPutty, de forma, a alterar as configurações dos equipamentos. No que respeita à linguagem Lua, esta foi escolhida por se apresentar como uma linguagem de *scripting* caracterizada como rápida, portátil e de elevado desempenho [65]. Ao nível arquitetural, os *scripts* de arranque são suportados por duas bibliotecas criadas com funções de configuração dos SBCs e BIG-IP. A criação destas bibliotecas teve como intuito diminuir os conhecimentos necessários para adaptar as configurações, conseguindo assim separar o conhecimento da linguagem Lua do conhecimento do equipamento. Através dos *scripts* de arranque, o utilizador só necessita de saber qual o elemento a alterar e que parâmetro alterar, sendo o resto dos comandos realizados pelas bibliotecas. Um caso de simples análise encontra-se presente na Figura 4.13 e Figura 4.14. Na Figura 4.13 são apresentadas as configurações no *script* que se pretendem alterar, sendo na Figura 4.14 representados os comandos que o *script* despoleta no equipamento.

```
backup="BACK_PTIN_SBC1_2013_05_09_pre.gz"
defaultGateway="default-gateway 172.22.12.121"
hostrouteID="111.122.122.116"
dnsCommands={"dest-network 172.22.12.26","description LabDNS"}


init()
restoreBackup(backup)
configuration()
systemConfig(defaultGateway)
hostRoute(hostrouteID, dnsCommands)
```

Figura 4.13: Script exemplo de cenário

```
login as: xxxxx
xxxx@yyy.yyy.yyy.yy's password:
sbc1ptin> enable
Password:
sbc1ptin# restore-backup-config BACK_PTIN_SBC1_2013_05_09_pre.gz
sbc1ptin# configure terminal
sbc1ptin(configure)# system
sbc1ptin(system)# system-config
sbc1ptin(system-config)# select
sbc1ptin(system-config)# default-gateway 172.22.12.121
sbc1ptin(system-config)# exit
sbc1ptin(system)# host-route
sbc1ptin(host-route)# select
1: 111.122.122.116 : 255.255.255.0 : 10.102.42.33

selection:1
sbc1ptin(host-route)# dest-network 172.22.12.26
sbc1ptin(host-route)# description LabDNS
sbc1ptin(host-route)# exit
sbc1ptin(host-route)# exit
sbc1ptin(system)# exit
sbc1ptin(configure)#
```

Figura 4.14: Comandos associados à execução do *script* exemplo de cenário

4.4 Processo de criação dos testes

O processo de criação dos testes segue as etapas apresentadas na Figura 4.15, abrangendo todo o processo de criação desde a análise dos requisitos ate à inserção das validações nos *scripts* SIPp a executar. O processo inicia-se com a passagem dos requisitos para testes executáveis. Neste analisam-se os requisitos da solução e dividem-se os mesmos em testes executáveis capazes de ser automatizados. O processo levado a cabo na realização da passagem de requisitos para testes é apresentado no apêndice B.

Figura 4.15: Procedimento para a criação de testes

Com o caderno de testes definido e fechado, é necessário executar todos os testes na rede onde o equipamento se encontra implementado de forma a obter evidências dos seus comportamentos. Uma vez executados os testes, é necessário filtrar as capturas de rede, armazenando

somente a informação associada ao teste que se validou. Com o termino da captura e filtragem das evidências, tem-se a solução preparada para iniciar a criação dos *scripts* SIPp.

O processo de criação dos *scripts* SIPp é realizado de forma automática tirando proveito de *scripts* Perl *open source* disponibilizados online. A decisão de automatizar o processo de geração dos *scripts* teve como principal justificação o tamanho avultado que as mensagens SIP apresentam tornando o processo manual propenso a erros. Em termos funcionais, através do processo do automático de criação dos *scripts* SIPp pretende-se criar uma solução capaz de analisar mensagens similares à presente na Figura 4.16 e gerar mensagens SIPp idênticas à presente na Figura 4.17.

```
ACK sip:emergency-117@10.111.71.73:9070;transport=udp SIP/2.0
To: <sip:117@lab11a.sec.ims.telecom.pt>;tag=153064393
From:
testessbcbtip.internet.geo1<sip:testessbcbtip.internet.geo1@lab11a.sec.ims.telecom.pt>;tag
=4d0fb453
Via: SIP/2.0/UDP 10.112.209.178:5061;branch=z9hG4bK-d87543-753284826-1--d87543-;rport
Call-ID: 5a5aa72d5a276e0b
CSeq: 1 ACK
Contact: <sip:testessbcbtip.internet.geo1@10.112.209.178:5061>
Max-Forwards: 70
User-Agent: eyeBeam release 3003x stamp 16296 (sn:7f0d7c702cfc9e9a7977)
Content-Length: 0
```

Figura 4.16: Conteúdo de ficheiro pcap capturado na rede do cliente

```
<send>
  <![CDATA[
    ACK sip:emergency-[field0]@10.111.71.73:9070;transport=udp SIP/2.0
    To: <sip:[field0]@[field1]> [peer_tag_param]
    From: [field2]<sip:[field2]@[field1]>;tag=[pid]SIPpTag00[call_number]
    Via: SIP/2.0/UDP [local_ip]:[local_port];branch=[branch];rport
    Call-ID: [call_id]
    CSeq: 1 ACK
    Contact: <sip:[field2]@[local_ip]:[local_port]>
    Max-Forwards: 70
    User-Agent: eyeBeam release 3003x stamp 16296 (sn:7f0d7c702cfc9e9a7977)
    Content-Length: [len]
  ]]>
</send>
```

Figura 4.17: Mensagem SIPp criada através do conteúdo do ficheiro pcap

Das aplicações online capazes de gerar *scripts* de forma automática através de evidências de rede destacam-se duas:

- pcap2sipp
- sniff2sipp

Após análise comparativa das funcionalidades de ambas (análise presente no Apêndice C), optou-se pela utilização da aplicação sniff2sipp, havendo contudo necessidade de a adaptar de forma a cumprir as necessidades da solução proposta. A questão mais problemática passa pelo

não suporte do cabeçalho 802.1Q, que se encontra presente em todas as capturas de tráfego, tornando a sua análise essencial.

Após a criação dos scripts é necessário configurar validações de forma a verificar o comportamento associado aos testes. As verificações dos testes são realizadas através de expressões regulares responsáveis por analisar determinados cabeçalhos e verificar se os mesmos possuem os valores que seriam expectáveis. Em termos funcionais, as expressões regulares necessárias para os testes podem ser de três tipos distintos: expressões regulares de presença de cabeçalhos, expressões regulares de validação de cabeçalhos estáticos e expressão regulares de cabeçalhos dinâmicos. Através das expressões regulares de presença de cabeçalhos valida-se a existência de determinados cabeçalhos na mensagem. As expressões regulares de valores fixos, operam com valores normalmente estáticos. Um caso prático encontra-se presente na validação do RURI das chamadas de emergência (valores aceitáveis são 112, 115 ou 117). Por último, as validações dinâmicas servem para validar valores e campos que apresentam um maior dinamismo. Um caso prático deste tipo de validações são as validações do *user part* em todos os cabeçalhos de identificação de utilizadores.

A explicação das expressões regulares dos três tipos indicados é apresentada em maior detalhe no Apêndice D, onde se apresentam *templates* para a geração fácil das validações.

4.4.1 sniff2Sipp

O sniff2sipp tem como intuito a geração de *scripts* SIPp através da análise direta do tráfego na rede ou de capturas de tráfego previamente realizadas. Através das capturas, o sniff2sipp desfragmenta as *frames* e com a informação SIP gera os *scripts* SIPp. O modo como a ferramenta opera agiliza o processo de geração de *scripts* SIPp. Contudo a aplicação, por defeito considera que após o cabeçalho Ethernet é apresentado o cabeçalho IP, não pondo a hipótese de existência de VLANs. A não análise do cabeçalho 802.1Q torna inviável a passagem das capturas realizadas na rede para *scripts*.

Para colocar a solução a operar foi necessário adaptar a aplicação de forma a ser verificada a presença de cabeçalhos 802.1Q antes de analisar o IP. O tratamento das VLANS é realizado na classe Reassemble.pm, classe auxiliar da ferramenta sniff2sipp com funções de decomposição e junção de pacotes. Este tratamento é efetuado após a análise dos cabeçalhos da camada dois IP e antes de iniciar a análise dos cabeçalhos da camada três. A verificação da presença dos cabeçalhos é efetuada verificando se o cabeçalho seguinte ao Ethernet é IPv4 ou IPv6. Caso

não seja, significa que existe um protocolo intermédio entre a camada Ethernet e IP. No caso de existir um protocolo intermédio é necessário fazer a análise do cabeçalho 802.1Q. Ao nível da implementação, o código implementado é similar ao apresentado na Figura 4.18. Inicialmente verifica-se qual o protocolo seguinte ao protocolo Ethernet presente na *frame*. Caso o protocolo seja IP, o processamento da *frame* segue o fluxo original da aplicação analisando o IP, protocolo de transporte e protocolo SIP. Caso o protocolo não seja IP é necessário recolher a *frame*, retirar o cabeçalho 802.1Q e preparar o *payload* de forma a garantir que quando é realizada a análise do protocolo IP é o cabeçalho IP que se encontra no início da *frame*.

```
my $aux= unpack('C', $packet) >>4;
 if($aux != 4 && $aux !=6) {
    $packet= _read802qpkt($packet);
 }

sub _read802qpkt ($) {

    my $packet = shift;

    my $vlan = Net::Packet::VLAN->new(raw => $packet);

    my $payload = substr($vlan->raw, 4);

    return $payload;
}
```

Figura 4.18: Código de tratamento de VLANs

Juntamente com as alterações realizadas ao nível do tratamento de VLANs, alterou-se também o modo como o sniff2sipp manipula valores como a identificação dos utilizadores e os domínios nos diferentes cabeçalhos SIP. Por defeito o sniff2sipp tira proveito das variáveis [remote_ip], [remote_port] e [service] para a dinamização dos valores das capturas. Estas variáveis alteram o comportamento dos cenários substituindo domínios por informação incorretas como o IP e porta. Não sendo esse o comportamento expectável, optou-se por eliminar em grande parte o uso de variáveis internas definidas pelo SIPp, substituindo-as por variáveis auxiliares do tipo [field*n*].

O comportamento por defeito da ferramenta SIPp é similar ao apresentado na Figura 4.19. Sendo recolhida a informação do domínio e substituída sempre pelo IP e porta de origem.

```
$sip_string =~ s/(from):\s*([^:]+):([^@]+)@[^>\s]+/1: $2:$3\@[local_ip]:[local_port]/i;
```

Figura 4.19: Comportamento por defeito da aplicação sniff2sipp

De forma a utilizar variáveis [field*n*] acrescentou-se um novo argumento à aplicação sniff2sipp

(–v) para indicação dos valores que se pretende ter como dinâmicos. Os valores associados ao argumento –v são considerados valores a dinamizar, sendo inseridos num ficheiro auxiliar ao *script* e substituídos por variáveis do tipo [field*n*]. O código para a substituição dos valores é similar ao apresentado na Figura 4.20, sendo em cada linha da mensagem SIP verificada a presença das variáveis definidas em -v.

```
foreach my $val (@variables) {
  $sip_string =~ s/$val/[field$count]/gi;
  $count++;
}
```

Figura 4.20: Substituição dos valores definidos no -v por valores field*n*

4.5 Metodologia de execução dos testes

A execução dos testes tem quatro fases distintas como apresentado na Figura 4.21. No decorrer das diferentes fases obtêm-se os testes para execução, executa-se a bateria de testes, capturam-se evidências e por fim gera-se o relatório.

Os testes necessitam de estar sempre atualizados e acessíveis para quem necessitar de realizar as validações. Com esta premissa em mente decidiu-se disponibilizar os *scripts* SIPp num repositório Subversion (SVN), tornando os testes acessíveis a quem necessita de realizar validações. Após fazer *checkout* dos testes do repositório SVN para máquina local, o restante processo de execução da bateria de testes é automático, estando ao encargo da aplicação desenvolvida em Java de nome SIPp-ATester.

A aplicação desenvolvida transfere os ficheiros de teste obtidos do repositório SVN para as máquinas virtuais com capacidade de executar os *scripts* SIPp. Executa a bateria de testes guardando evidências sobre o comportamento de cada um dos testes e gera relatórios de execução dos cenários, apresentando informações percentuais sobre a execução dos testes, bem como detalhes sobre cada teste executado.

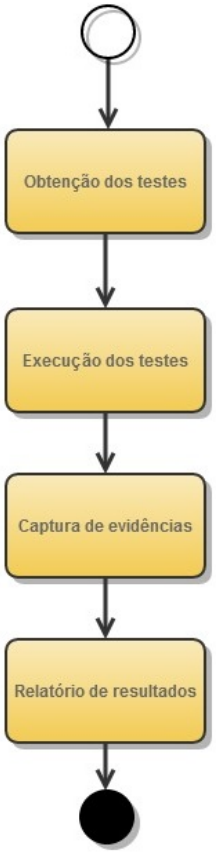


Figura 4.21: Procedimento para a execução de testes

4.5.1 SIPp-ATester

A aplicação SIP-ATester apresenta funções em todas as etapas do procedimento de execução de testes. Na fase de obtenção dos testes, a aplicação SIPp-ATester é responsável por transferir os ficheiros de testes para as máquinas virtuais através de ligações File Transfer Protocol (FTP). Na etapa de execução de testes, a aplicação envia informação para as máquinas sobre que testes executar e de que forma os executar, indicando informação de conetividades, temporizadores, entre outras. Relativamente às evidências, a aplicação captura-as em paralelo com a execução dos testes. Em relação ao relatório dos testes, o mesmo é gerado pela aplicação na fase final da execução dos testes, indicando percentagem de testes com sucesso e insucesso, bem como informação detalhada sobre cada teste.

Ao nível arquitetural, a aplicação é composta por cinco blocos distintos, como apresentado na Figura 4.22.

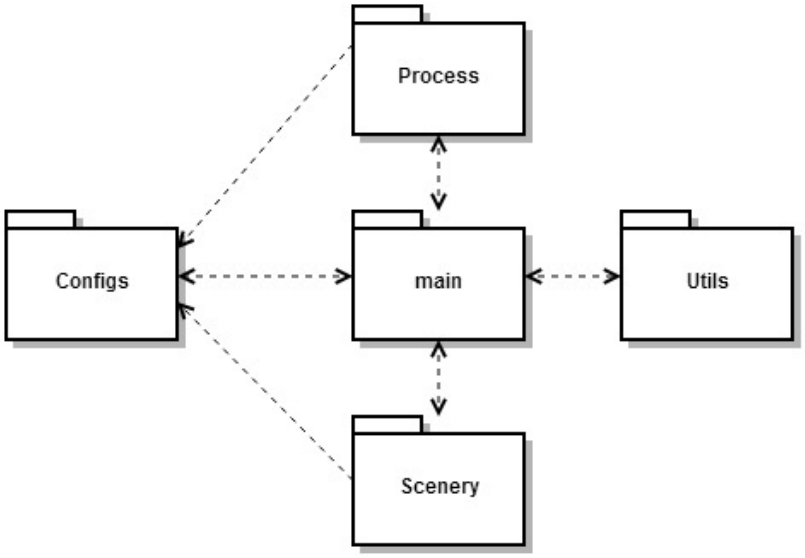


Figura 4.22: Diagrama de blocos da aplicação SIPp-ATester

O bloco de controlo denominado de main, possui as funções de agregação dos diferentes módulos, decorrendo no main o fluxo de execução. O bloco Configs faz a leitura dos ficheiros de configurações obtendo informação sobre as máquinas virtuais, os cenários a testar e os testes a executar. O bloco Process, organiza os ficheiros nas máquinas virtuais e gera ficheiros *Shell script* de execução dos *scripts* SIPp. O bloco Scenario indica os argumentos a executar sobre os *scripts* criados pelo bloco Process. O bloco Utils contém todas as funcionalidades auxiliares para a execução com sucesso dos testes, como clientes FTP e SSH.

A criação de *Shell scripts* para execução dos *scripts* SIPp é necessária uma vez que, os *scripts* SIPp representam um teste, mas consoante o cenário existe um conjunto de pré-condições a cumprir, sendo estas precondições garantidas pelos *Shell scripts* de execução dos *scripts* SIPp.

No que respeita ao fluxo da aplicação, este é similar ao apresentado na Figura 4.23.

Após o utilizador iniciar a aplicação são obtidas as configurações para realização dos testes e configurações dos cenários a testar. Com as configurações em memória são estabelecidas sessões SSH e FTP com as máquinas virtuais para a execução dos comandos e transferência dos *scripts* SIPp. Após passagem para as máquinas virtuais de toda a informação necessária para a execução dos diferentes testes dá-se início à execução dos testes. No final da execução da bateria de testes é gerado o relatório de resultados.

Ao nível de configurações, a aplicação SIPp-ATester não apresenta uma interface iterativa de

comunicação legando a realização das configurações para os dois ficheiros de suporte. O ficheiro app.config possui as configurações necessárias para executar os testes enquanto, o ficheiro de CSV tem a informação dos testes a executar. O ficheiro app.config apresenta um formato extenso tendo um conjunto de configurações de preparação de ambiente. Estas configurações são detalhadas na Tabela 4.1, na Tabela 4.2 e na Tabela 4.3.

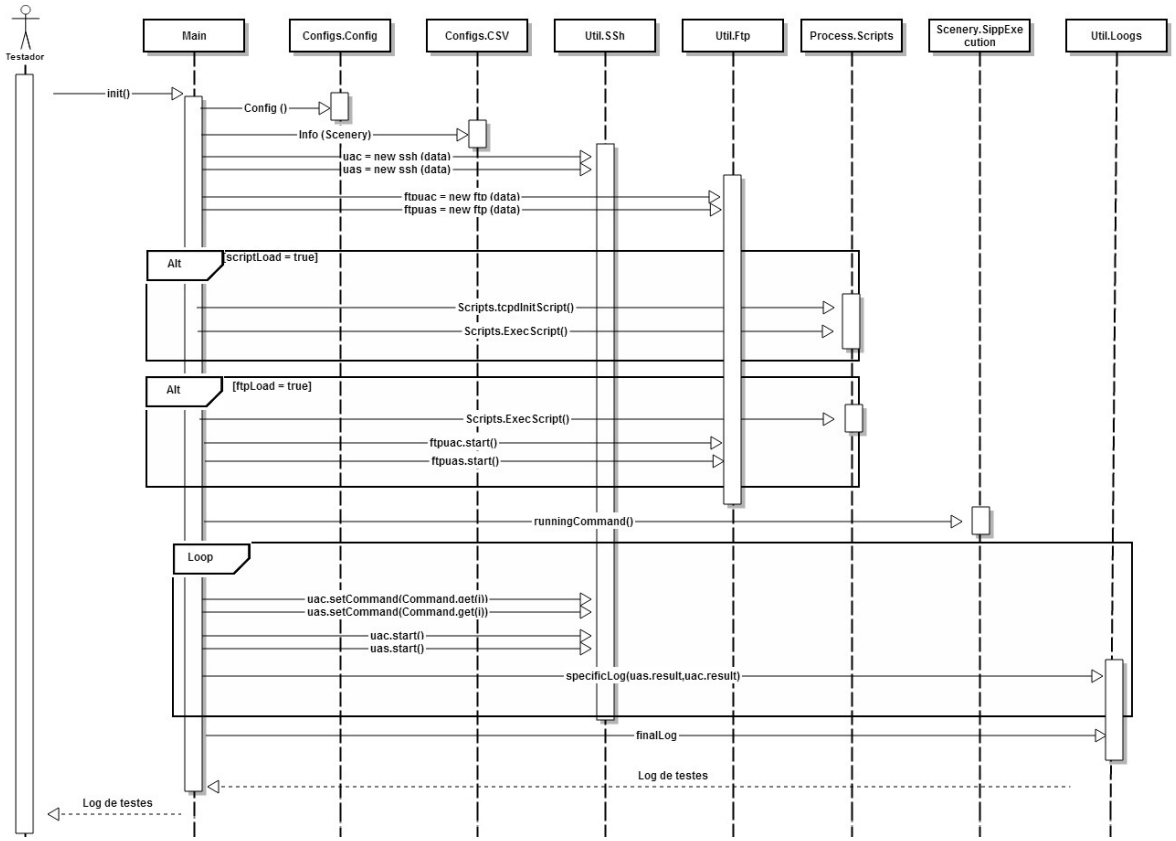


Figura 4.23: Diagrama de sequência da aplicação SIPp-ATester

No que respeita a informação de autenticação e acesso a máquinas esta é identificada pelos campos presentes na Tabela 4.2

Campo	Descrição
test.version	Indica a versão a validar, servindo para estruturação da informação
test.scenario	Cenário a testar, este pode apresentar diferentes valores, sendo usados para a criação de *scripts* SIPp melhorados para algumas soluções
test.scenario.sub	Grupo associado ao cenário representativo de um conjunto de testes específicos
test.folder	Indicação da pasta a criar nas máquinas remotas
test.input.file	Identificação dos testes a executar
test.script.load	Informação booleana indicativa da necessidade de criação dos *Shell Script* para execução dos testes SIPp
test.ftp.load	Informação booleana indicativa da necessidade de transferência dos testes para as máquinas remotas
uas.folder	Indicação da pasta onde os testes de UAS se encontram
uac.folder	Indicação da pasta onde os testes de UAC se encontram

Tabela 4.1: Informação do cenário a testar

Campo	Descrição
uas.user	Nome do utilizador a usar na máquina remota a operar como UAS
uas.host	Endereço de O&M da máquina remota a operar como UAS
uas.pass	Password na máquina remota a operar como UAS
uac.user	Nome do utilizador a usar na máquina remota a operar como UAC
uac.host	Endereço de O&M da máquina remota a operar como UAC
uac.pass	Password na máquina remota a operar como UAC

Tabela 4.2: Informação de autenticação de UAC e UAS

Para além da informação das máquinas existe um conjunto de parâmetros que necessitam de ser identificados para os *scripts* operarem tanto para o UAS como para o UAC (demonstrados somente os valores de UAC).

Campo	Descrição
uac.sipp.addr	Endereço IP local usado no SIPp
uac.sipp.port	Porto local usado no SIPp
uac.sipp.gateway.addr	Endereço IP remoto usado no SIPp
uac.sipp.gateway.port	Porto remoto usado no SIPp
uac.sipp.second.gateway.addr	Endereço IP remoto secundário, cenários específicos SIPp
uac.sipp.second.gateway.port	Porto remoto secundário, cenários específicos SIPp
uac.sipp.interface	Interface de captura do teste SIPp
uas.cap.timer	Timeout para finalização da captura

Tabela 4.3: Parâmetros usados no SIPp

Após as configurações do cénario estarem em memória é necessário carregar a informação do ficheiro test.input.file indicado anteriormente. Neste ficheiro encontra-se a informação dos testes a executar indicando também como estes se relacionam. Para cada teste é indicado os *scripts* a usar e os ficheiros auxiliares, caso existam. Por norma, os testes são constituídos por ficheiros de UAC e UAS, no entanto, podem existir testes somente com o UAC ou só com o UAS.

Relativamente aos testes executados, estes podem ser de diferentes tipos apresentando diferentes papéis no fluxo normal dos testes. Podendo os testes ser de registo (r), de anulação de registo (d) ou testes de chamadas (c). A separação dos testes em diferentes tipos deve-se ao facto dos testes de chamadas, por norma, terem de ser realizados por utilizadores registados, o que leva à necessidade de fazer o registo do utilizador em cada teste executado.

Toda a informação presente no ficheiro test.input.file é usada como parâmetros para executar o *Shell scripts* de validação dos *scripts* SIPp. Os *Shell scripts* gerados para a execução dos SIPp são similares ao apresentado na Figura 4.24. Estes, são usados para invocar os diferentes *scripts* SIPp presentes no fluxo de cada teste, tirando proveito da linguagem *bash* presente nos sistemas Linux.

```
#!/bin/bash
#Call sipp and execute

  if [ $2 == "-" ]
  then
    ./sipp -sf $6 -inf $7 -i $3 -p $4 -m 1 $5 &> empty
    sleep 1
    ./sipp -sf $1 -i $3 -p $4 -m 1 $5 -recv_timeout 5000 &> empty
     echo $?
     sleep 1
     ./sipp -sf $8 -inf $9 -i $3 -p $4 -m 1 $5 &> empty
  else
    ./sipp -sf $6 -inf $7 -i $3 -p $4 -m 1 $5 &> empty
    sleep 1
    ./sipp -sf $1 -inf $2 -i $3 -p $4 -m 1 $5 -recv_timeout 5000 &> empty
    echo $?
    sleep 1
    ./sipp -sf $8 -inf $9 -i $3 -p $4 -m 1 $5 &> empty
  fi
```

Figura 4.24: Script tipo usado para executar os scripts SIPp

Os *scripts* de execução do SIPp são denominados de SippExec, sendo executados para a validação de todos os testes a realizar. Para executar o SippExec é usado um conjunto alargado de parâmetros como apresentado na Figura 4.25.

Relativamente a resultados, o SippExec retorna o *exit code* associado ao SIPp executado. A necessidade de retornar o *exit code*, em detrimento do retorno normal do SIPp, verifica-se porque a ferramenta SIPp é uma aplicação iterativa que opera sobre uma *Shell Unix*, não sendo possível

```
./SippExec sips/6.4m4p4/Sc_Sec/SBC_CLF_146_008-03.xml sips/6.4m4p4/Sc_Sec/SBC_CLF_146_008-
03 88.214.177.129 5060 88.214.177.134:5060 sips/6.4m4p4/Sc_Sec/SBC_CLF_111_002-01.xml
sips/6.4m4p4/Sc_Sec/SBC_CLF_111_002-01 sips/6.4m4p4/Sc_Sec/SBC_CLF_111_001-01.xml
sips/6.4m4p4/Sc Sec/SBC CLF 111 001-01
```

Figura 4.25: Comando usado para a execução do Shell script

obter o resultado do teste de forma direta. De forma a resolver a problemática verifica-se a necessidade de obter um valor estático associado ao resultado dos testes executados, recorrendo-se para tal à análise dos *exit codes*.

Os comandos, aplicações e utilitários executados em ambiente Unix possuem um *exit code* associado, não sendo a ferramenta SIPp uma exceção. Através do *exit code* torna-se possível a verificação do comportamento do teste em cada uma das máquinas de forma simples e rápida. Se o valor do *exit code* for igual a zero, significa que o teste teve sucesso sendo todos os restantes valores indicativos de insucesso do teste.

4.6 Resumo

Ao longo do capítulo é apresentada a proposta de automatização de testes desenvolvida, sendo dado enfâse aos procedimentos e aplicações desenvolvidos neste âmbito. Com a proposta apresentada pretende-se endereçar todas as questões relevantes para a automatização de testes em equipamentos de rede IMS, criando uma solução útil e com valor. No que diz respeito aos equipamentos suportados, embora a solução tenha sido preparada e testada para equipamentos específicos, considera-se que a mesma se encontra preparada para dar serviço e auxiliar no processo de automatização de diferentes equipamentos.

Capítulo 5

Teste e validação da solução proposta

Neste capítulo aborda-se o funcionamento da arquitetura definida (secção 5.1), descrevendo passo-a-passo o processo de validação de um cenário IMS. De seguida são apresentadas as mais-valias que a solução apresenta comparativamente com as validações manuais (secção 5.2). Por fim são apresentadas as aplicações que as diferentes partes da solução possuem.

5.1 Demonstração de funcionalidades

Devido à solução se encontrar a ser usada para validação de equipamentos e configurações presentes na rede da PT Inovação, não se pode apresentar exemplos reais dos testes realizados. De forma a colmatar a indisponibilidade dos dados reais, são realizadas demonstrações com dados simulados, descrevendo as diferentes fases inseridas na solução, demonstrando comportamentos e funcionalidades.

Para a demonstrar a integração e aplicabilidade dos procedimentos e das aplicações usadas, demonstra-se todo o processo desde a realização dos testes em rede cliente até à obtenção dos resultados da execução de testes.

5.1.1 Criação dos testes

A primeira tarefa definida na criação de testes é a adaptação de requisitos para testes (Apêndice B), tendo no final desta tarefa uma bateria de testes preparada para executar na rede do cliente. A execução dos testes na rede do cliente tem por intuito obter capturas idênticas à presente

na Figura 5.1, sendo necessário realizar filtragens para que a captura se possa apresentar desta forma.

```
UAC.IP -> SBC.FE.IP SIP/SDP 1013 Request: INVITE sip:00351753@domain.pt, with session description
SBC.FE.IP -> UAC.IP SIP 370 Status: 100 Trying
SBC.BE.IP-> UAS.IP SIP/SDP 1298 Request: INVITE sip:753@domain.pt;user=phone, with session description
UAS.IP -> SBC.BE.IPSIP 348 Status: 100 Trying
UAS.IP -> SBC.BE.IPSIP 765 Status: 180 Ringing
SBC.FE.IP -> UAC.IP SIP 518 Status: 180 Ringing
UAS.IP -> SBC.BE.IPSIP/SDP 1246 Status: 200 OK, with session description
SBC.FE.IP -> UAC.IP SIP/SDP 998 Status: 200 OK, with session description
UAC.IP -> SBC.FE.IP SIP 559 Request: ACK sip:00351753@10.11.111.20:5060;transport=udp
SBC.BE.IP-> UAS.IP SIP 728 Request: ACK sip:753@10.11.112.45:5060
UAC.IP -> SBC.FE.IP SIP 636 Request: BYE sip:00351753@10.11.111.20:5060;transport=udp
SBC.BE.IP-> UAS.IP SIP 940 Request: BYE sip:753@10.11.112.45:5060
UAS.IP -> SBC.BE.IPSIP 423 Status: 200 OK
SBC.FE.IP -> UAC.IP SIP 395 Status: 200 OK
```

Figura 5.1: Fluxo de mensagens trocadas entre o UAC e UAS sobre o equipamento a validar

As mensagens presentes na Figura 5.1 geram dois *scripts* SIPp distintos operando um como cliente e outro como servidor. Após obtenção e filtragem das capturas da rede do cliente é necessário gerar os *scripts* a executar no SIPp, estando a geração dos mesmos a cargo da aplicação sniff2sipp. Para iniciar o sniff2sipp é necessário executar um comando similar ao presente na Figura 5.2, indicando o nome do pcap a analisar, os portos sobre os quais o SIP opera e quais os parâmetros que se pretende colocar como dinâmicos [field*n*].

```
./sniff2sipp.pl -f SBC_CLF_121_007-01.pcapng -p 5050-5070 -v
753;domain.pt;10.11.111.20;10.11.112.45
```

Figura 5.2: Comando perl para execução da aplicação sniff2sipp

Após a execução do comando da Figura 5.2 é apresentada a informação indicativa sobre o sucesso ou não da execução dos testes, como ilustrado na Figura 5.3, sendo demonstrados todos os ficheiros criados. Relativamente aos *scripts* gerados, estes apresentam-se em tudo similares as capturas de rede como é possível verificar na Figura 5.4 e Figura 5.5. Após a criação dos *scripts* SIPp aplicam-se as expressões regulares seguindo o *template* descrito no Apêndice D, sendo este o último passo na geração dos testes.

```
Wrote scenario file SBC.CLF/SBC_CLF_121_007-01:10.83.84.18.xml
Wrote scenario file SBC.CLF/SBC_CLF_121_007-01:10.11.112.45.xml
Wrote scenario file SBC.CLF/SBC_CLF_121_007-01:10.12.32.34.xml
Wrote scenario file SBC.CLF/SBC_CLF_121_007-01:10.11.111.20.xml
Wrote audio file 10.83.84.18:11994.pcap
Wrote audio file 10.11.112.45:10182.pcap
Wrote audio file 10.12.32.34:10178.pcap
Wrote audio file 10.11.111.20:11996.pcap
```

Figura 5.3: Output da aplicação sniff2sipp

```
INVITE sip:00351753@domain.pt SIP/2.0
 To: <sip:00351753@domain.pt>
 From: testessbc <sip:testessbc@domain.pt>;tag=e77fb60b
 Via: SIP/2.0/UDP 10.11.111.20:5060;branch=z9hG4bK-d87543-459565262-1--d87543-
;rport
 Call-ID: 557db6214157c532
 CSeq: 1 INVITE
 Contact: <sip:testessbc@10.11.111.20:5060>
 Max-Forwards: 70
 Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE
 Content-Type: application/sdp
 User-Agent: eyeBeam release 3003x stamp 16296 (sn:7f0d7c702cfc9e9a7977)
 Content-Length: 359

 v=0
 o=- 8969140 8969159 IN IP4 10.11.111.20
 s=eyeBeam
 c=IN IP4 10.11.111.20
 t=0 0
 m=audio 10178 RTP/AVP 100 8 3 18 98 101
 a=alt:1 1 : 24C46F31 4C9BC3D7 10.11.111.20 10178
 a=fmtp:101 0-15
 a=rtpmap:100 speex/16000
 a=rtpmap:8 pcma/8000
 a=rtpmap:3 gsm/8000
 a=rtpmap:18 g729/8000
 a=rtpmap:98 ilbc/8000
 a=rtpmap:101 telephone-event/8000
 a=sendrecv
```

Figura 5.4: Conteúdo da mensagem INVITE presente na captura de rede do cliente

```
<send >
    <![CDATA[
      INVITE sip:00351[field0]@[field1] SIP/2.0
      To: <sip:00351[field0]@[field1]>
      From: [field2]<sip:[field2]@[field1]>;tag=[pid]SIPpTag00[call_number]
      Via: SIP/2.0/UDP [local_ip]:[local_port];branch=[branch];rport
      Call-ID: [call_id]
      CSeq: 1 INVITE
      Contact: <sip:[field2]@[local_ip]:[local_port]>
      Max-Forwards: 70
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE
      User-Agent: eyeBeam release 3003x stamp 16296 (sn:7f0d7c702cfc9e9a7977)
      Content-Type: application/sdp
      Content-Length: [len]

      v=0
      o=- 8969140 8969159 IN IP4 [local_ip]
      s=eyeBeam
      c=IN IP4 [media_ip]
      t=0 0
      m=audio [auto_media_port] RTP/AVP 100 8 3 18 98 101
      a=alt:1 1 : 24C46F31 4C9BC3D7 [field3] 10178
      a=fmtp:101 0-15
      a=rtpmap:100 speex/16000
      a=rtpmap:8 pcma/8000
      a=rtpmap:3 gsm/8000
      a=rtpmap:18 g729/8000
      a=rtpmap:98 ilbc/8000
      a=rtpmap:101 telephone-event/8000
      a=sendrecv
    ]]>
  </send>
```

Figura 5.5: *Script* SIPp gerado através da captura da rede do cliente

5.1.2 Criação do cenário

A criação do cenário inicia-se com a verificação da presença das configurações ao nível de rede. Estas podem ser validadas com a realização de um comando *ping* verificando o estado das conectividades IP. O passo seguinte consiste em validar as configurações ao nível do serviço, sendo para tal usada a aplicação SIP-Scenario, onde se indica que acesso se pretende testar (VLAN, IP, Porto) e se obtém um relatório idêntico ao apresentado na Figura 5.6.

```
################################
System-Config

INFO: The system default gateway is 192.168.10.254
################################
Network-Interface

INFO: The Network Interface M00:3456 respond to ICMPs on the IP 10.11.1.45
################################
Realm-Config

INFO: The realm emp_back_sig has the realm emp_back_media as parent realm
INFO: The realm emp_emergency_back_sig has the realm emp_back_media as parent realm
################################
SipInterface

INFO: The realm emp_back_sig has the IP:Port 10.11.1.1:5060 enabled
INFO: The realm emp_emergency_back_sig has the IP:Port 10.11.1.1:7062 enabled
################################
Session Agents

INFO: The Session-Agent pcscf.domain.pt is enabled
       The SBC expect responses to OPTIONS method
INFO: The Session-Agent pcscf.domain.pt is enabled
       The SBC expect responses to OPTIONS method
################################
Local Policy

Info: There's no Local Policy configured on these realm
################################
Steering Pool

INFO: The Steering Pool configured on the IP 10.11.1.50 starts on port 10000 and ends
on 19999 using the realm emp_back_media
################################
AccessControl

INFO: The realm emp_back_sig has an ACL from IP 10.11.1.1 to *
INFO: The realm emp_emergency_back_sig has an ACL from IP 10.11.1.1 to *
```

Figura 5.6: Relatório do estado das configurações obtido pelo SIP-Scenario

Com o levantamento das configurações realizado é necessário adaptar o *script* de arranque caso o mesmo não esteja atualizado, sendo de seguida o *script* de arranque executado. A execução do *script* realiza modificações no equipamento fazendo o mapeamento direto entre as configurações descritas e as realizadas como é visível na Figura 5.7.

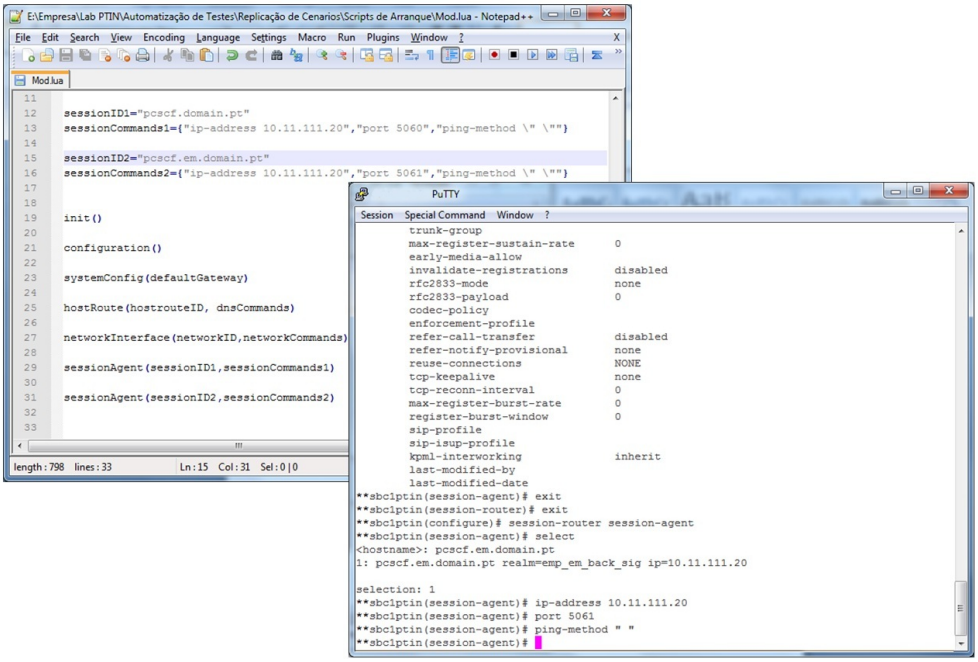


Figura 5.7: Script de arranque e comandos executados no equipamento

Com a realização das configurações do equipamento, fica-se com o ambiente da proposta de automatização de testes preparado podendo ser iniciada a execução dos testes.

5.1.3 Execução dos testes

A execução da bateria de testes é o último procedimento a realizar para conseguir validar o cenário. Com o ficheiro de configuração preparado, deve ser executada a aplicação SIPp-ATester e analisar a informação disponibilizada.

```
Configuration loaded...
Tests loaded...
Server Connected...
Client Connected...
Generating Scripts ...
Scripts generated
Test 1/124
UAC test named: SBC_CLF_111_002-01.xml
UAS test named: SBC_CLF_111_002-01.xml
UAC result:0
UAS result:0
Test took 11020 milliseconds
Test 2/124
UAC test named: SBC_CLF_121_006-01.xml
UAS test named: SBC_CLF_121_006-01.xml
UAC result:0
UAS result:0
Test took 11029 milliseconds
```

Figura 5.8: Output intermédio da aplicação SIPp-ATester

Através da análise intermédia apresentada na Figura 5.8 verifica-se que a execução dos testes demora aproximadamente dez segundos. Com o final da execução da bateria testes é disponibilizado o relatório final da aplicação SIPp-ATester, tal como é ilustrado na Figura 5.9.

```
Registo de execução de testes
-----------------------------
Versão: MOD
Data: 2013-7-05 15:35:0

Resumo
------
1) Testes com sucesso    :123 de 124 (99.0%)
2) Testes com insucesso  :1 de 124 (0.0%)
2.1) Expressão regulares :1 de 124 (0.0%)
2.2) Questões de rede    :0 de 124 (0.0%)


Detalhes
--------
Teste SBC_CLF_111_002-01.xml:
uac value: 0
uas value: 0
```

Figura 5.9: Relatório final da aplicação SIPp-ATester

Analisando a Figura 5.9 verifica-se a existência de um teste com insucesso associado à validação do conteúdo do cabeçalho P-Preferred-Identity como se verifica na Figura 5.10. Na Figura 5.10 verifica-se que no *script* é aplicada uma expressão regular ao cabeçalho P-Preferred-Identity, sendo capturado o URI associado ao cabeçalho. Este valor é de seguida comparado com o valor presente na variável [field0] de modo a verificar se o valor recebido é o expectável.

```
<recv request="INVITE" crlf="true">
  <action>
  <ereg regexp="([^<]+)(<[^ ]+)"
  search_in="hdr"
  header="P-Preferred-Identity:"
  check_it="true"
  assign_to="all,all,ppi">
  </ereg>

  <log message="Info capture: P-Preferred-Identity='[$ppi]'"></log>

  <assignstr assign_to="1" value="[field0]" />
  <assignstr assign_to="2" value="[$ppi]" />
  <log message="PPI_File='[$2]'">
  </log>

  <strcmp assign_to="result" variable="1" variable2="2" />
  <log message="Variable1='[$1]' Variable2='[$2]' Result=[$result]">
  </log>

  <ereg regexp="[-+]?([0-9]*\.[0-9]+|[0-9]+)" search_in="var" variable="result"
assign_to="result" check_it_inverse="true" />
 </action>
  </recv>
```

Figura 5.10: Expressão regular aplicada ao teste SIPp falhado

Ao ser executado o teste, é gerada uma captura de rede onde se pode analisar a causa do teste falhado. No exemplo em questão, observa-se na Figura 5.11, que a mensagem SIP não possui o cabeçalho PPI presente e, neste caso, o problema terá que ser corrigido no equipamento alvo de teste.

```
INVITE sip:753@10.11.111.20:5060 SIP/2.0
 Via: SIP/2.0/UDP 10.102.42.14:5060;branch=z9hG4bKiamu7u107gtgiostr4l1.1
 To: <sip:753@domain.pt>;tag=13451SIPpTag011
 From: testessbc <sip:testessbc@domain.pt>;tag=18058SIPpTag001
 Call-ID: 1-18058@88.214.177.129
 CSeq: 3 INVITE
 Contact: <sip:testessbc-bsltqk1st7e7e@10.11.111.20:5060;transport=udp>
 Max-Forwards: 69
 Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE
 User-Agent: eyeBeam release 3003x stamp 16296 (sn:7f0d7c702cfc9e9a7977)
 Content-Type: application/sdp
 Content-Length: 309
 Route: <sip:3Zqkv7%1Bbaqnqaaaadd0BFAJS4aaqaaaaaisip%3Atestessbc%domain.pt;lr>
 P-Media-Release: hngl5rpaqatuiidpvqktjvurr8ibif5vof8dgfvf010np3ms6o4aerahh31anvt50000g01
 P-Access-Network-Info: 3GPP-GERAN;cgi-3gpp
```

Figura 5.11: Evidência do teste falhado

5.2 Resultados obtidos e aplicabilidade

Em termos funcionais, a solução apresenta-se prática para a validação de um conjunto de cenários e equipamentos. Comparativamente com o método utilizado anteriormente (método de validação manual) a solução de automatização é realmente vantajosa, conseguindo diminuir tempos de cerca de uma hora para poucos segundos. Embora o esforço levado a cabo para o desenvolvimento da solução de automatização tenha sido de grande consumo de recursos durante cerca de quatro meses, este é compensado pela diminuição de tempos que a solução causa nas diferentes validações que são realizadas no decorrer de um ano. Durante um ano em média são realizadas três a quatro validações de *firmware* dos SBCs na rede do cliente, tendo em cada uma das validações um conjunto de testes extenso. Através da automatização de testes consegue-se realizar o mesmo número de validações de *firmware*, tendo cada uma, um muito menor tempo de execução.

De forma a comparar o tempo gasto na realização de testes manuais com o tempo gasto na realização de testes automáticos, recorreu-se à plataforma de contabilização de tempos da PT Inovação. Através da plataforma verificou-se o tempo gasto nas últimas validações de *firmware* fazendo uma análise comparativa entre as validações manuais e automáticas. A última validação feita de forma manual foi realizada por um engenheiro sénior com vasta experiência em soluções IMS e com conhecimentos avançados em SIP e SBCs. Nesta validação foram realizados cerca de vinte e cinco testes sobre a rede do cliente de forma remota, tendo demorado em cada teste em média entre 40 a 60 minutos. Neste tempo estão englobadas as tarefas de preparação dos terminais, execução dos testes, captura de evidências, filtragem das capturas, análise dos fluxos, análise dos cabeçalhos e indicação do resultado do teste. Relativamente aos testes automáticos, como demonstrado anteriormente, em média demoram cerca de dez segundos a fazer as funções que se encontram associadas aos testes manuais, verificando-se melhorias significativas na solução automática.

Para além da solução de automatização ser de rápida execução por parte do utilizador, é também de fácil aprendizagem. Um engenheiro júnior realizou validações com a solução de automatização proposta, com tempos de validação muito próximos dos gastos pelo desenvolvedor. Os valores são apresentados no gráfico demonstrado na Figura 5.12.

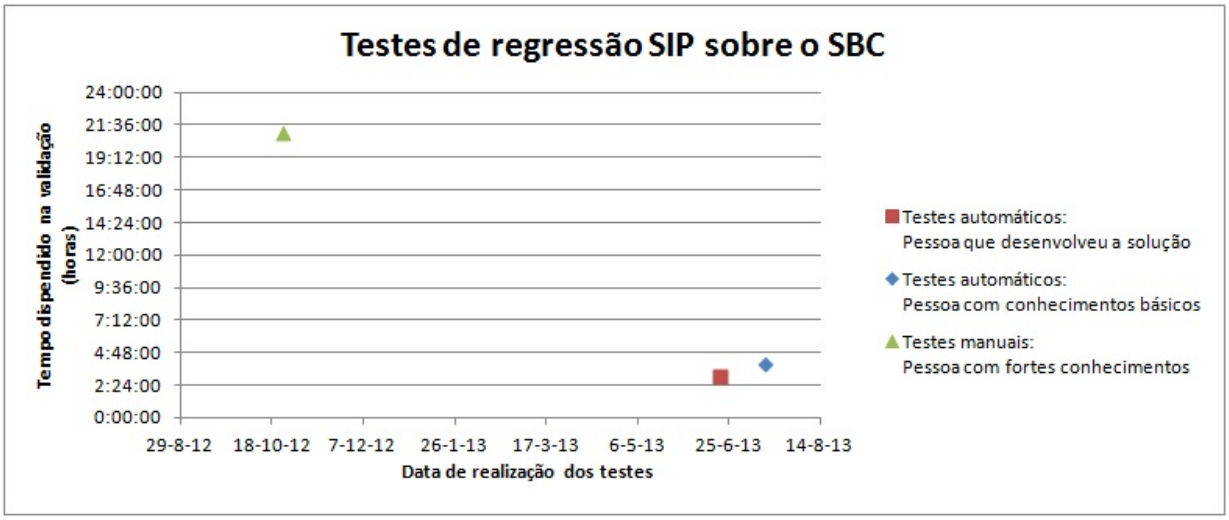


Figura 5.12: Gráfico de tempos despendidos em validações de firmware

Juntamente com a redução de tempos verificam-se também vantagens associadas com o controlo de erros e cobertura de testes realizados. Na Figura 5.12 os testes realizados de forma manual foram cerca de vinte e cinco enquanto, na solução automática foram cerca de trezentos tendo demorado mesmo assim um tempo muito inferior (aproximadamente duas horas e trinta minutos para a realização dos trezentos testes). No que diz respeito à confiabilidade, como os testes automáticos trabalham com validações booleanas os resultados são determinísticos, havendo um maior grau de confiabilidade nos resultados obtidos.

Em termos de aplicabilidade todos os procedimentos desenvolvidos podem ser executados num conjunto distinto de validações. Podendo ser aplicados a testes de regressão para a execução das baterias de testes, executados para validação de testes específicos para validação de situações de erros, usados para validação de testes de sanidade, entre outos. Tendo sido já aplicados em cenários de laboratório e cenários de produção. Os procedimentos desenvolvidos podem ser aplicados como um todo para a validação das diferentes soluções, como podem ser aplicados separadamente para a execução de diferentes cenários. A execução da replicação de cenários permite a criação do ambiente do cliente em ambiente de desenvolvimento para a implementação de novas soluções e teste das mesmas. A criação de testes possibilita a replicação de cenários

anómalos para análise do problema e resolução do mesmo. A execução de testes auxilia na validação de testes de rede de forma mais expedita validando fluxos.

5.3 Resumo

Ao longo do capítulo apresentou-se o funcionamento da arquitetura definida, demonstrado passo-a-passo o processo de automatização e validação de uma solução de rede IMS. Foram também apresentadas as mais-valias da utilização da plataforma em detrimento dos testes realizados de forma manual e as aplicações que a solução dispõe.

Capítulo 6

Conclusão

Com o trabalho realizado pretendeu-se desenvolver uma solução capaz de agilizar o processo de validação de equipamentos IMS, diminuindo o esforço e os recursos necessários para a validação das soluções IMS no decorrer do seu período de vida.

Como resultado do trabalho apresentou-se uma proposta para a validação de diferentes equipamentos da rede IMS a operar sobre o protocolo SIP. Na proposta são endereçadas todas as questões desde a criação dos testes até à execução dos mesmos, sendo a proposta genérica o suficiente para ser adaptada para um conjunto de equipamentos distintos. De forma a validar a proposta implementou-se e validou-se a mesma em equipamentos de entrada da rede sobre diferentes cenários em que os equipamentos se integram.

O desenvolvimento e implementação da proposta de automatização de testes cumprem os objetivos delineados para o trabalho, tendo a proposta de validação otimizado o trabalho de validação de testes, tornando possível a realização de um número de testes mais elevado num espaço de tempo mais curto, conseguindo ao mesmo tempo aumentar o grau de confiabilidade nas validações.

Juntamente com o cumprimento dos objetivos, o trabalho serviu para o desenvolvimento de conhecimentos em diversas áreas das redes de nova geração, alargando os conhecimentos tidos de arquiteturas, protocolos e soluções de rede através do estudo e análise de arquiteturas como o IMS. O processo de validação automática foi criado após estudo e análise dos processos realizados na validação manual, tornando-se necessário analisar fluxos e as mensagens nos diferentes equipamentos de forma a conseguir criar uma solução funcional.

No que respeita às dificuldades encontradas no trabalho, estas foram de diferentes níveis de-

vido à abrangência que se pretendia que a solução tivesse. Inicialmente foi necessário analisar o processo de validação manual e verificar que tarefas poderiam ser executadas de forma automática e como estas podiam ser automatizadas. Esta tarefa levou à necessidade de criar um ambiente capaz de emular tráfego SIP, sendo necessário estudar ferramentas de emulação, configurações de equipamentos a validar, entre outros. Uma vez concebida a proposta, foi necessário criar, adaptar e utilizar aplicações em diversas linguagens, sobre diferentes ambientes. Juntamente com as dificuldades técnicas associadas à proposta existiram dificuldades de usabilidade. Neste contexto a proposta necessitou de ser desenvolvida de forma a ser de fácil utilização para diferentes pessoas usufruírem da mesma garantindo que todo o processo era intuitivo.

Embora as dificuldades ao longo do projeto tenham sido diversas, considera-se que as mesmas foram ultrapassadas com sucesso, estando a solução a ser usada pela PT Inovação para a validação dos seus SBCs em diferentes cenários de teste.

Apesar de a solução apresentada ser uma proposta, esta inclui um conjunto de ferramentas e ideias capazes de auxiliar no processo de validação de um conjunto alargado de equipamentos, apresentando-se a dissertação com uma fonte de informação agregada sobre problemáticas e soluções associadas à validação de testes SIP.

6.1 Trabalho Futuro

Embora os objetivos definidos tenham sido atingidos e a solução de automatização de testes esteja a operar para SIP existem trabalhos e melhorias a realizar. Pretende-se melhorar as capacidades de algumas aplicações automatizando alguns processos em falta (ex. inserção de expressões regulares).

Para além das melhorias nas aplicações criadas e modificadas, pretende-se alargar os protocolos suportados, incluindo por exemplo, o protocolo Diameter. Para validar o protocolo Diameter sobre a solução proposta é necessário emular tráfego e gerar *scripts* para a ferramenta de emulação de forma automática. A ferramenta de emulação já foi estudada, sendo o Seagull a ferramenta a usar. A aplicação para geração automática dos *scripts* necessita de ser criada de raiz, visto não haver bibliotecas ou aplicações capazes de analisar capturas de rede Diameter.

Appendix A

Revista Saber & Fazer - Automatização de testes em Redes IMS

Automatização de testes SIP em redes IMS

Autores: José Carlos Silva, David Gonçalves, António Amaral

Sumário Executivo

A qualidade de um projeto encontra-se diretamente relacionada com a capacidade dos testes abrangerem todos os requisitos existentes, garantindo-se que quanto mais testes forem efetuados, menos problemas serão reportados.

A lista de requisitos de um projeto pode ser densa e complexa, havendo a necessidade de os decompor em requisitos mais simples para os poder mapear em testes específicos. O tempo e os recursos gastos para a execução dos testes, dependem claramente do método usado para a implementação deste processo. A automatização do processo de testes apresenta-se como um conceito chave, já que permite melhorar o tempo de execução dos testes e minimizar os erros de análise dos resultados obtidos.

Os testes para validação de novas versões de *firmware*, fundamentais para garantir que todas as funcionalidades (novas e antigas) são garantidas e que os problemas são corrigidos, se forem automatizados resultam num ganho de tempo significativo, face à validação dos mesmos usando um processo manual.

No entanto, embora as vantagens adjacentes à utilização de testes automáticos sejam evidentes, o processo de mapeamento dos requisitos para testes é complexo, porque a informação simples de analisar por um humano por vezes tem de ser decomposta em elementos com maior detalhe que possam ser analisados por máquinas.

Baseado na experiencia prática de validação do *firmware* dos SBCs (*Session Border Controller*) [1] de soluções implementadas em rede de operador, este artigo apresenta uma metodologia de desenvolvimento capaz de, através da análise dos requisitos, mapeá-los em testes automáticos, diminuindo desta forma o esforço contínuo que é necessário para realizar entregas recorrentes aos clientes.

Palavras-Chave: automatização de testes, IMS, SIP

Introdução

As redes de telecomunicações encontram-se em constante evolução, impulsionadas pela constante procura de soluções cada vez mais ricas e convergentes que por sua vez têm vindo a contribuir para a mudança de paradigma no desenvolvimento dos próprios serviços, bem como para a mudança para uma arquitetura de rede *IP Multimedia Subsystem* (IMS) [2].

A arquitetura IMS, caracteriza-se por ser uma arquitetura funcional estratificada e que tem por princípio a disponibilização de serviços do tipo *session-based* a terminais servidos por redes de acesso distintas [3]. Esta estratificação permite a criação de um sistema interoperável, capaz de disponibilizar um conjunto vasto de serviços, à custa da garantia dos requisitos de qualidade de serviço e considerando sempre as questões de mobilidade. A arquitetura apresenta-se decomposta em três níveis: transporte, controlo, serviço.

Ao nível do transporte, são endereçadas todas as questões de conectividades com as diferentes redes de acesso, permitindo a sua dissociação com os serviços disponibilizados [4]. Ao nível do controlo, trata toda a sinalização associada ao estabelecimento de sessões e são consideradas as questões relacionadas com a autenticação e taxação. Finalmente ao nível do serviço, encontra-se a disponibilização de serviços avançados sobre a rede IMS, sendo este nível responsável pela disponibilização de serviços de redes legadas (serviços disponibilizados pelo CAMEL [5] nas redes de segunda-geração) ou a disponibilização de serviços inovadores desenvolvidos de raiz para as redes IMS.

Caracterizando-se como sendo uma arquitetura modular capaz de disponibilizar um conjunto extenso de serviços, as redes IMS apresentam-se como o passo natural na evolução das redes dos operadores e já o começam a ser atualmente, permitindo-lhes desenvolver soluções com requisitos e funcionalidades distintas, indo sempre de encontro às necessidades dos utilizadores.

A necessidade de validar os requisitos na sua totalidade apresenta-se como um fator fundamental para a determinação do sucesso ou insucesso de uma solução. Este princípio, está associado a qualquer tipo de produto/serviço, onde se enquadram todas as soluções em redes IMS. Contudo, a realização de validações da totalidade dos requisitos de forma manual caracteriza-se por ser processo moroso, de elevado consumo de recursos humanos e suscetível a erros. A automatização das validações apresenta-se, assim, como o caminho indicado para a obtenção de uma solução robusta e capaz de ter em consideração todos os requisitos.

Sendo o protocolo SIP (*Session Initiation Protocol*) o protocolo de sinalização usado em redes IMS e sendo este protocolo composto por diferentes tipos de mensagens, com diferentes *headers*, é fundamental executar processos de validação da integração dos componentes que implementam este protocolo na rede IMS. Este ponto assume particular importância quando se está perante elementos de rede responsáveis por adaptações protocolares de *headers* SIP, como é o caso do SBC, que é o elemento de entrada SIP na rede IMS do operador, fazendo a interligação entre a rede de acesso - onde está o cliente - e a rede do operador.

Motivação

Os testes a realizar para validar uma solução estão diretamente relacionados com o que se pretende validar e com a maturidade da solução implementada. Desta forma, pode-se separar o âmbito dos testes em testes funcionais, testes de sanidade, testes de aceitação e testes de regressão.

Os testes funcionais têm como objetivo validar o comportamento funcional da solução ou componente em causa. Os testes de sanidade pretendem validar se a solução apresenta um nível de maturidade para ser submetida a testes mais detalhados com um nível de profundidade maior. Os testes de aceitação pretendem validar todos os casos num ambiente o mais próximo do ambiente de produção e, finalmente, os testes de regressão são executados em ambiente de produção, de forma a confirmar que a alteração introduzida na solução (quer seja por atualização de *hardware*, quer seja por atualização de *software*) não tem efeitos colaterais que possam comprometer as funcionalidades existentes.

Como se verifica, existe um conjunto extenso de pontos de validação de requisitos, sendo que, para cada um deles, dever-se-á realizar os testes que mapeiam os requisitos na sua totalidade ou um subconjunto destes. O que se pretende com a solução de automatização de testes SIP em redes IMS é definir procedimentos para a validação das soluções suportadas num determinado equipamento, sendo neste caso particular consideradas as soluções suportadas no SBC.

Na operacionalização da solução automática de testes SIP foi necessário isolar o SBC da rede e ter o controlo dos *inputs* e *outputs* que condicionam o seu comportamento. Posteriormente, definiu-se um conjunto de requisitos sujeitos a validação automática e foi necessário desenvolver mecanismos capazes de executar os testes e realizar capturas ao nível SIP, para posteriormente se fazer a análise dos resultados finais. Tendo controlo sobre os *inputs* e *outputs*, torna-se possível simular o ambiente real aplicado apenas ao SBC, já que se consegue replicar os fluxos de sessões SIP iguais aos que se tem na rede do operador, à custa de ferramentas de emulação de tráfego responsáveis por executar cenários de validação. Neste caso particular, foi usado o SIPp [6], que permite a geração de tráfego SIP. Esta ferramenta caracteriza-se por apresentar um bom nível de maturidade ao nível SIP e por estar muito bem documentada com exemplos que caracterizam diferentes fluxos de chamadas SIP, o que simplificou a sua utilização.

A operacionalização dos fluxos de testes carregados na ferramenta SIPp, se for feita de forma manual, torna-se num processo lento e sujeito a erros. Por esse motivo, utilizou-se um processo automático para que, com base em capturas de redes reais, fossem operacionalizados os casos de testes para os cenários pretendidos.

No processo de execução dos testes foi necessário ter em consideração que os cenários de envio e receção devem ser executados de forma simultânea e devem ter a capacidade de registar as evidências respetivas.

Desenho de solução

A complexidade do processo de automatização de testes não permite que a validação seja realizada através de uma aplicação unitária que atua de forma completamente independente do utilizador. Através de procedimentos normalizados, consegue-se automatizar a grande maioria dos processos de validação com resultados bastante satisfatórios quando comparados com o processo de validação manual existente.

Operacionalização dos cenários

Para a operacionalização dos cenários é seguido o fluxo descrito na Figura 1, sendo considerados os seguintes aspetos:

- Configurações de rede
- Verificação das configurações de serviço
- Implementação das configurações

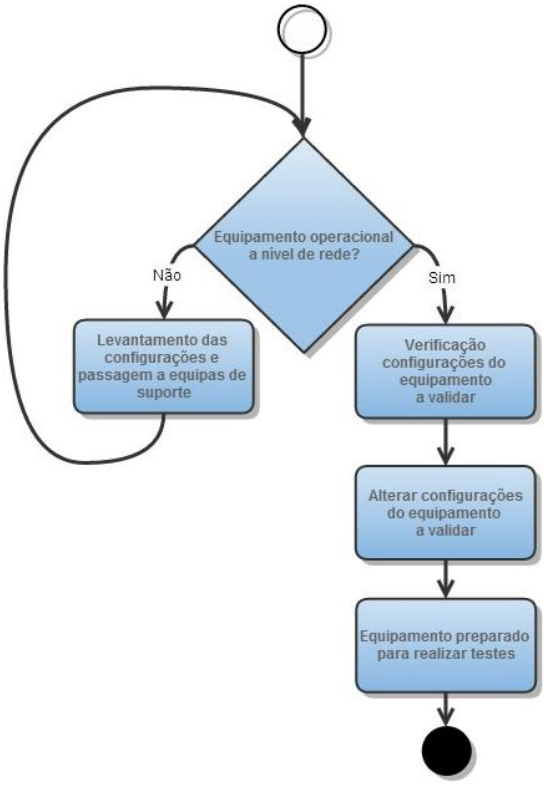


Figura 1 - Fluxo de operacionalização dos cenários

Configurações de rede

De forma a ter um cenário igual ao do ambiente de produção, operacionalizou-se toda a rede envolvente para replicar o cenário real onde o SBC opera, garantindo-se que todo o endereçamento do IP (*Internet Protocol*) e das VLANs (*Virtual Local Access Networks*) é igual ao do ambiente de produção.

A grande vantagem desta abordagem passa por facilitar a importação das configurações e execução dos testes sobre as condições implementadas na rede de produção, sem haver necessidade de envolver os próprios SBCs que estão a dar o serviço real.

Verificação das configurações

Antes da realização das alterações nas configurações do equipamento é necessário fazer o levantamento das configurações que necessitam de ser modificadas. Estas verificações encontram-se fortemente associadas com as configurações do equipamento, sendo necessário possuir conhecimento razoável do equipamento para realizar o levantamento das configurações que necessitam de sofrer alterações de modo a disponibilizarem serviço. No caso específico de validação de soluções no SBC, foi necessário separar funcionalidades tais como desativação dos cenários de registo quando se realizam testes de estabelecimento de sessão, desativação do envio das mensagens *keep alive* para não ter testes com entropia, entre outros. Para facilitar a validação, desenvolveu-se uma aplicação simples capaz de realizar de forma rápida o levantamento das configurações a alterar.

Implementação das configurações

Após ter o equipamento preparado na rede e ter sido feito o levantamento das configurações, é necessário proceder à sua implementação. Para tal, são usados *scripts* aplicados a clientes SSH (*Secure Shell*) e ou Telnet com as configurações a atualizar. A ferramenta de SSH/Telnet escolhida para a realização das configurações foi o ExtraPutty [7], que se liga ao equipamento e executa um conjunto de comandos com as alterações requeridas. Para cada cenário a validar é necessário um *script* de arranque (*script* associado ao ExtraPutty) distinto, de modo a habilitar as configurações do equipamento.

Criação de testes

No processo de criação de testes foram consideradas duas fases: a criação de *scripts* SIPp e a aplicação de expressões regulares.

Criação dos *scripts* SIPp

Para a criação dos *scripts* SIPp implementou-se um processo automático que usa a ferramenta *open source* sniff2sipp [8] para gerar os *scripts,* tendo como *input* as capturas de rede no formato ".pcap". Contudo, para a ferramenta operar devidamente, foi necessário adaptar a análise da pilha protocolar usada pela ferramenta, dando a possibilidade de existirem cabeçalhos com identificação de VLANs e tornando, assim, possível analisar a totalidade das capturas realizadas em ambiente de produção. Para além da alteração da pilha protocolar suportada, foi dada a possibilidade de inserir valores dinâmicos nos *scripts* SIPp através da definição de variáveis.

Aplicação de expressões regulares

A criação dos cenários, por si só, não permite a realização da totalidade das validações, já que têm que ser feitas com um nível de detalhe aprofundado, consequência da necessária validação de todo o conteúdo das mensagens SIP. Usando a possibilidade do SIPp suportar expressões regulares, definiu-se um conjunto de expressões regulares aplicáveis a cada cenário.

A utilização das expressões regulares para validação dos conteúdos dos cabeçalhos das mensagens SIP possibilita a realização de um vasto conjunto de análises, já que podem ser considerados vários parâmetros. De seguida, apresenta-se um exemplo de uma expressão

regular aplicada a uma mensagem SIP INVITE, onde é analisado o conteúdo do *header* From, comparando o valor do *header* com o valor recebido na mensagem anterior (Figura 3).

```
<recv request="INVITE" crlf="true">
 <action>
   <ereg regexp=".*" search_in="hdr" header="From:" assign_to="1"
   </ereg>

   <log message="From is [last_From]. Custom header is [$1]"/>
  </log>
 </action>
</recv>
```

Figura 2 - Exemplo de uma expressão regular em SIPp

Em todos os testes é necessário indicar a expressão regular que se pretende validar, indicando o valor ou o conjunto de valores que seriam expectáveis receber, que excerto da mensagem se pretende analisar, se é pretendido analisar a mensagem na sua totalidade "msg" ou se é apenas sobre um cabeçalho específico "hmr" e se é pretendido que se armazenem em variáveis os valores pretendidos.

Se o resultado das expressões regulares for positivo, então o teste é dado como passado. Caso contrário, o teste é dado como falhado e é indicado qual o motivo para o insucesso do mesmo.

Execução de testes

O procedimento de execução de testes engloba todas as atividades associadas com a execução, verificação dos comportamentos, forma de realizar os testes e garantia de registo de evidências.

Na execução dos testes foram considerados os atores apresentados na figura seguinte, cada um deles implementado em máquinas distintas.

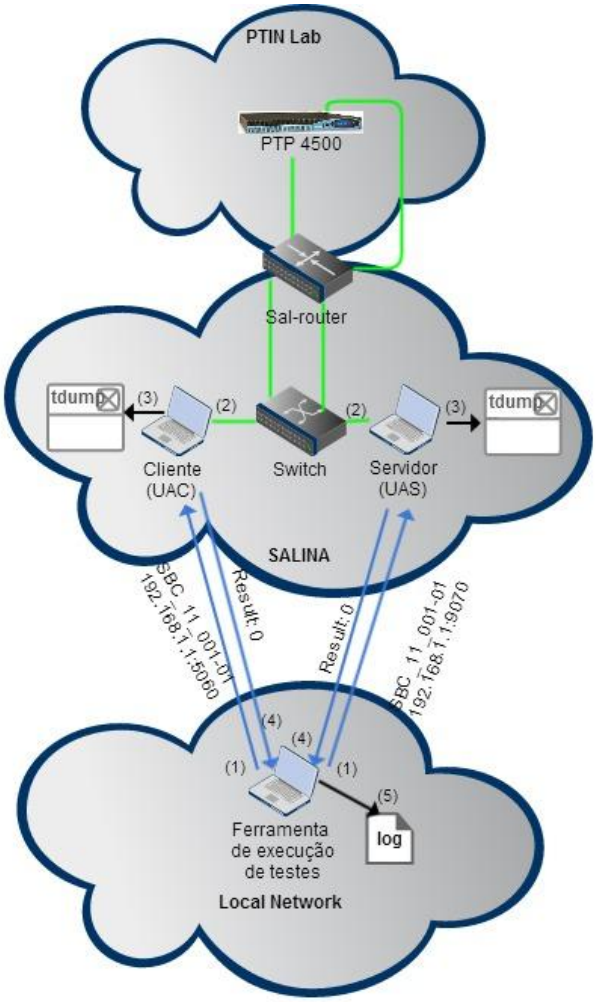


Figura 3 - Diagrama Lógico de execução de testes

A ferramenta de execução de testes é o ator responsável por iniciar a execução da bateria de testes, indicando tanto ao cliente como ao servidor quais os *scripts* SIPp que devem ser executados. Os testes a validar são executados de forma síncrona entre o elemento responsável pela geração do *input,* fazendo este o papel de *User Agent Client* (UAC), e o elemento que recebe o *output* gerado pelo equipamento a validar, sendo este denominado de *User Agent Server* (UAS). De forma a realizar a validação, pode-se optar por duas soluções distintas para a implementação da ferramenta de execução de testes. Por um lado, pode-se desenvolver uma aplicação de controlo de máquinas sobre ligações remotas para a execução dos testes. Por outro lado, podem-se integrar os testes numa plataforma de testes automáticos (ex. Cucumber) capaz de executar um conjunto de testes controlando a plataforma e as máquinas UAC e UAS. No caso específico da validação dos SBCs, optou-se pelo desenvolvimento de raiz de uma aplicação de controlo das máquinas, não tendo sido possível fazer a integração das validações numa plataforma de validação automática.

O registo de evidências pode ser obtido de formas distintas dependendo da plataforma escolhida para a realização da validação de testes. Através da plataforma Cucumber, possui-se acesso a um conjunto de *reports* detalhados sobre o funcionamento do teste. Ao nível

aplicacional obtém-se evidências através da integração de comandos *tcpdump* na execução dos *scripts* SIPp, bem como através de *reports* globais da bateria de testes executados.

O *report* global de execução da bateria de testes apresenta-se na Figura 5, onde se mostra informação sobre a bateria de testes, estatísticas sobre as validações e informação detalhada da execução de cada teste individual.

```
Registo de execução de testes
-----------------------------
Versão:
Data:

Resumo
------
1) Testes com sucesso    :
2) Testes com insucesso  :
2.1) Expressão regulares :
2.2) Questões de Rede    :


Detalhes
--------
```

Figura 4 – Relatório de execução de testes

A informação da bateria de testes indica o estado dos testes e a data da execução. A informação estatística apresenta os resultados, de forma agregada, da execução da bateria de testes, a percentagem de testes com sucesso e a percentagem de testes com insucesso. A informação detalhada indica para cada teste o output obtido pela ferramenta de validação associado à execução do SIPp, tanto do lado do cliente como do lado do servidor.

Nas evidências obtidas pelo *tcpdump* é possível analisar a informação do fluxo associado ao *User Agent* (UA). A Figura 5 apresenta um exemplo da informação capturada.

```
192.168.7.13 -> 192.168.8.15 SIP 668 Request: REGISTER sip:dom.ims.pt    (fetch bindings)
192.168.8.15 -> 192.168.7.13 SIP 566 Status: 401 Unauthorized    (0 bindings)
192.168.7.13 -> 192.168.8.15 SIP 921 Request: REGISTER sip: dom.ims.pt    t    (fetch bindings)
192.168.8.15 -> 192.168.7.13 SIP 691 Status: 200 OK    (0 bindings)

192.168.7.13 -> 192.168.8.15 SIP/SDP 949 Request: INVITE tel:123456789, with session description
192.168.8.15 -> 192.168.7.13 SIP 330 Status: 100 Trying
192.168.8.15 -> 192.168.7.13 SIP 481 Status: 180 Ringing
192.168.8.15 -> 192.168.7.13 SIP/SDP 965 Status: 200 OK, with session description
192.168.7.13 -> 192.168.8.15 SIP 495 Request: ACK tel:123456789
192.168.7.13 -> 192.168.8.15 SIP 572 Request: BYE tel:123456789
192.168.8.15 -> 192.168.7.13 SIP 363 Status: 200 OK

192.168.7.13 -> 192.168.8.15 SIP 665 Request: REGISTER sip: dom.ims.pt    (fetch bindings)
192.168.8.15 -> 192.168.7.13 SIP 566 Status: 401 Unauthorized    (0 bindings)
192.168.7.13 -> 192.168.8.15 SIP 918 Request: REGISTER sip: dom.ims.pt    (fetch bindings)
192.168.8.15 -> 192.168.7.13 SIP 618 Status: 200 OK    (0 bindings)
```

Figura 5 – Informação capturada na execução de testes automáticos

A captura apresenta apenas os fluxos existentes entre o SBC e a respetiva máquina UA onde a captura é realizada. Em relação ao fluxo da captura, só é constituído por mensagens associadas ao teste que se pretende validar, filtrando os restantes pacotes. Para cada teste são gerados dois ficheiros ".pcap", um com o comportamento de *input* do SBC e outro com o *output*.

Testes e resultados

Os resultados obtidos com o processo automático permitiram concluir um ganho efetivo de tempo no processo de execução dos testes de validação de *firmware* do SBC. Estes resultados assumem particular relevância quando se trata de um elemento de rede SIP responsável por um elevado conjunto de manipulações de diferentes *headers* em diferentes mensagens SIP e cuja análise do resultado dessas manipulações é extremamente morosa e sujeita a erros.

Num processo manual, para a validação do resultado dos testes é necessário configurar terminais, realizar o teste capturando evidências, filtrar as evidências para o teste em questão, fazer a análise do fluxo e de cabeçalhos específicos, descrever o resultado do teste, armazenar as evidências e analisar os resultados finais. Estes passos resultam em largos minutos desde que se inicia o processo de execução, até que se termina o teste com a respetiva análise. Conjuntos de testes anteriores para validação de versões de *firmware* usando o processo manual registam tempos médios de execução de um teste na ordem dos 60 minutos.

Nos mesmos testes de validação de *firmware*, mas usando o processo automático, registam-se tempos médios de execução de um teste na ordem dos 10 segundos, o que se traduz claramente num ganho bastante significativo de execução do teste. Neste período de tempo são executados os cenários com os números de terminal associados, são validados os fluxos (através do controlo restrito do SIPp sobre as mensagens que são esperadas receber ou não), são validados cabeçalhos específicos (através das expressões regulares, captura de evidências e lançamento de relatório de execução) e é registado o sucesso ou insucesso do teste.

O gráfico seguinte apresenta alguns resultados obtidos para o processo manual e para o processo automático. No processo manual, é apresentado o resultado da execução de 25 testes realizados por uma pessoa com larga experiência em SBCs e em redes IMS, conhecendo em detalhe o protocolo SIP e os requisitos associados a cada teste. No processo automático, são apresentados dois casos de execução de 300 testes, um deles executados por quem desenvolveu e implementou o processo de automatização e outro caso realizado por parte de uma pessoa que não participou no desenvolvimento do processo, ambos na sua fase de iniciação ao SBC e às redes IMS.

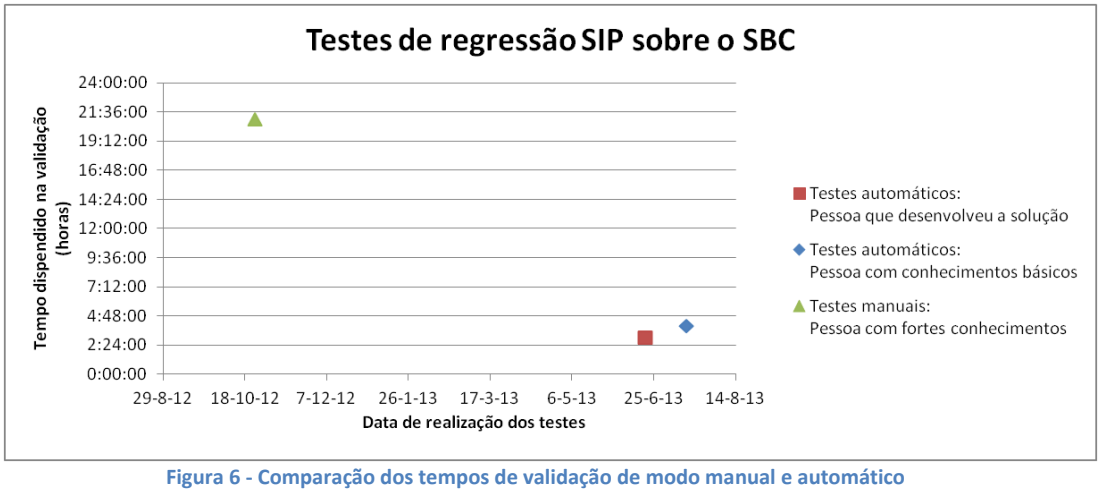


Figura 6 - Comparação dos tempos de validação de modo manual e automático

Analisando os resultados anteriores constata-se que, por um lado, com o processo automático consegue-se executar um maior número de testes, com uma redução significativa do tempo total de execução dos testes quando comparado com o tempo total de execução de um menor número de testes através do processo manual. Por outro lado, com o processo automático, a pessoa responsável pela execução dos testes não tem que ter um conhecimento aprofundado da tecnologia em questão, nem dos requisitos de cada testes, podendo esta tarefa de execução ser entregue às equipas responsáveis para execução dos testes de outras plataformas.

Conclusões

Ao ritmo acelerado a que as redes de telecomunicações evoluem, é fundamental diminuir tempos associados à realização de tarefas comuns, enquadrando-se nestas tarefas a validação de novas versões de *firmware* dos SBCs em redes IMS.

A consolidação do processo automático da validação de *firmware* do SBC requereu um esforço inicial significativo de forma a operacionalizar todo o processo. No entanto, este esforço inicial foi claramente compensado com o resultado final, já que com o processo implementado consegue-se, por um lado, executar um conjunto de testes mais alargado num período de tempo muito inferior e, por outro lado, é possível aumentar o número de validações efetuadas às mensagens SIP, à custa de uma diminuição de falhas de análise dos resultados obtidos, já que são usados comportamentos determinísticos no processo de análise.

O processo e métodos descritos para a validação automática SIP de versões de *firmware* ao nível do SBC podem ser aplicáveis a outros componentes das redes IMS que implementem a interface protocolar SIP, tais como o x-CSCF (*x-Call Session Control Function*), AS (*Application Server*), entre outros.

Finalmente importa referir que a solução de automatização de testes, para além do SIP, pode ser aplicada a outros protocolos (por exemplo o protocolo *diameter*), aplicando à solução geradores de tráfego que implementem esse protocolo.

Bibliography

[1] J. Hodges, *Session Border Controllers: Addressing Tomorrow's Requirements,* Heavy Reading, 2011.

[2] 3GPP, *IP Multimedia Subsystem (IMS); Stage 2,* 3rd Generation Partnership (3GPP), 2013.

[3] G. Camarillo e M. A. Garcia-Martin, The 3G IP multimedia subsystem IMS - merging the internet and the cellular worlds (2. ed.)., Wiley, 2006.

[4] Ericsson, *IMS – IP Multimedia Subsystem, The value of using the IMS architecture,* 2004.

[5] 3GPP, *Customized Applications for Mobile network Enhanced Logic (CAMEL) Phase X; Stage*

2, 3rd Generation Partnership (3GPP), 2010.

[6] R. Gayraud, “SIPp,” 2013. [Online]. Available: http://sipp.sourceforge.net/.

[7] ExtraPuTTY, “ExtraPuTTY,” 2013. [Online]. Available: http://www.extraputty.com/.

[8] T. Wilson, *sniff2sipp,* Digium, 2008.

Appendix B

Mapeamento de requisitos em testes: Caso prático

Para uma determinada solução é identificada a necessidade de garantir que as chamadas nacionais na rede funcionam sem a utilização do indicativo internacional. De forma a conseguir definir os testes para este é necessário realizar os seguintes passos:

- Identificar o conjunto de testes associados ao requisito.
- Descrever os cenários de teste e dos resultados expectáveis para cada um destes.

Requisito	Testes
Os RURI com números nacionais na rede não apresentam prefixo internacional	Chamada com o número 00351-111111111 Chamada com o número +351-111111111 Chamada com o número +354-111111111 Chamada com o número 111111111 Chamada com o número 351-111111111 Chamada com o número 00111111111

Tabela B.1: Testes associados com o requisito

Os testes acima indicados apresentam-se somente como um conjunto de testes a realizar, podendo ser definido um conjunto mais extenso de testes. Após estes estarem definidos, é necessário para cada um destes descrever o cenário do teste, indicar as ações a realizar e qual o resultado que se espera verificar no final deste.

Name		**Identification**
Chamada com prefixo internacional +351		SBC.21.001-01
Description		
Realização de uma chamada para um número em formato internacional, verificando se o +351 é retirado quando a chamada é passada para o core		
Initial Conditions		
N.A.		
Test Process		
	Actions	Expected Result:
01	Realização de uma chamada para o número +351yyyyyyyy	Verificar se o RURI no P-CSCF apresenta o formato yyyyyyyyy
Test post-conditions		
N.A.		
Additional Information		
Notes:		

Figura B.1: Configuração modelo de um teste

Appendix C

Análise comparativa entre pcap2sipp e sniff2sipp

O pcap2Sipp[66] é um projeto *open source* que se caracteriza-se como uma aplicação capaz de transformar um ficheiro pcap num *script* SIPp de forma a simular um cenário. Este apresenta-se simples sem a necessidade de possuir dependências de bibliotecas CPAN. As principais características do pcap2sipp apresentam-se na tabela C.1.

Vantagens	Desvantagens
Mobilidade, *script "Ready to use"*	Não suporta pacotes fragmentados
Rapidez de execução.	Não suporta outro protocolo sem ser UDP
Reduzido número de parâmetros	Não suporta IPv6
	Não suporta VLANs
	Estrutura rígida (Ethernet,IPv4,UDP,SIP)
	Analisa só um fluxo de cada vez
	Funciona só com ficheiros pcap
	Não gera ficheiro áudio

Tabela C.1: Vantagens e desvantagens do pcap2sipp

Em termos funcionais, o *script* Perl é relativamente simples, tendo como parâmetros o nome do pcap, os dois endereços do fluxo SIP e o IP local. Através desta informação o pcap2sipp gera os scripts de UAS e UAC, com as diferentes mensagens SIP da captura.

O sniff2sipp[67] apresenta-se como uma ferramenta usada para rápida geração de cenários SIPp através de capturas em tempo real ou através da análise de ficheiros pcap. Embora este seja um projeto de maior complexidade, é também um projeto com maiores potencialidades, tendo as

```
perl pcap2sipp.pl cfna_in.pcap 1.2.3.4 s/fhost:5060/[local_ip]:[local_port]/g
s/fhost/[local_ip]/g >cfna_in.xml
```

Figura C.1: Inicialização de *scripts* pcap2sipp

seguintes características:

Vantagens	Desvantagens
Suporte nativo de UDP, TCP	Dependências CPAN
Análise de múltiplos fluxos	Não suporta VLANs
Suporte para IPv6	Não lida com replicação de pacotes
Criação de ficheiros áudio	
Possibilidade de capturas em modo live	
Suporte de pacotes fragmentados	
Argumentos de arranque intuitivos	
Suporte de ficheiros pcapng	

Tabela C.2: Vantagens e desvantagens do sniff2sipp

Ao nível de execução, a ferramenta sniff2sipp apresenta como parâmetros –f que indica o ficheiro que se pretende analisar e o –p para indicação do conjunto de portas a analisar.

```
sniff2sipp -f SBC_111_111.pcapng -p 5000-10000
```

Figura C.2: Inicialização de *scripts* sniff2sipp

Realizando testes sobre ambas aplicações verificou-se que existia nas duas questões que teriam de ser endereçadas de forma a colocar a aplicação a operar sobre as capturas de rede obtidas na rede do cliente. Contudo, a ferramenta sniff2sipp apresenta-se como uma opção mais viável para a geração de cenários, endereçando questões de fragmentação de pacotes e suporte de múltiplos fluxos SIP simultâneos.

Appendix D

Template Expressões regulares

De forma a tornar o processo de validação de mensagens mais simples, definiu-se três templates de expressões regulares para os scripts SIPp.

D.1 Validação da existência do cabeçalho

Existe um conjunto alargado de testes onde o que se pretende validar é a existência ou não de um determinado cabeçalho. Este tipo de testes normalmente encontram-se associados com verificação da presença de informação que não deve ser passada para o *core*, ou para redes distintas tendo esta de ser eliminada no equipamento de fronteira. No que diz respeito à expressão regular construída, esta é constituída pela captura do valor associado ao cabeçalho a validar, comparação do valor e análise do resultado da comparação.

Inicialmente, sobre a mensagem que se pretende fazer a validação indica-se qual o cabeçalho que se pretende validar e armazena-se o seu valor como apresentado na Figura D.1.

```
<ereg regexp="(.*)"
   search_in="hdr"
   header="P-Preferred-Identity:"
   check_it="true"
   assign_to="ppi">
 </ereg>
```

Figura D.1: Captura do valores associados ao P-Preferred-Identity

Após obtenção do valor associado ao, neste caso o P-Preferred-Identity, é necessário verificar se este é igual a vazio (Figura D.2).

```
<assignstr assign_to="emptyValue" value="" />
<assignstr assign_to="ppiValue" value="[$ppi]" />

<strcmp assign_to="result" variable=" emptyValue" variable2="ppiValue" />
```

Figura D.2: Comparação dos valores do P-Preferred-Identity

Caso o resultado da comparação seja zero verifica-se que o valor do cabeçalho é nulo, sendo este o valor esperado. Caso o resultado seja um valor diferente de zero, verifica-se a presença do cabeçalho levando o teste a falhar. De forma a validar se o teste falhou, tira-se proveito da expressão regular presente na Figura D.3, onde valores diferentes de zero despoletam erros.

```
<ereg regexp="[-+]?([0-9]*\.[0-9]+|[0-9]+)" search_in="var" variable="result"
assign_to="result2" check_it_inverse="true" />
```

Figura D.3: Validação do valor obtido da comparação dos valores do P-Preferred-Identity

D.2 Validação de valores estáticos

Nas mensagens SIP em cenários específicos verificam-se valores fixos ou raramente modificações. Nestes casos, as validações podem ser simplificadas não necessitando de usar variáveis dinâmicas. Casos como a validação do RURI das chamadas de emergência ou chamadas em anónimo podem ser validados desta forma. Para usar este tipo de validações implementa-se expressões similares à presente na Figura D.4.

```
<ereg regexp="INVITE sip:11[2,5,7]@"
   search_in="msg"
   check_it="true"
   assign_to="all,all">
</ereg>
```

Figura D.4: Validação dos valores de emergência no RURI do INVITE

Nas validações feitas sobre o RURI é necessário analisar a mensagem na sua totalidade, devido à não existência de um cabeçalho associado. Nestes casos é necessário inserir o método que se esta a validar dentro da expressão regular, de forma à expressão a validar ser somente aplicada na linha do método. No caso da validação ser sobre um cabeçalho específico não existe a necessidade de inserção de informação suplementar à expressão que se pretende validar.

D.3 Validação de valores dinâmicos

Para a validação de valores com grande dinamismo recorre-se à utilização de ficheiros auxiliares juntamente com as expressões regulares. Para tal, compara-se as expressões regulares com os valores obtidos dos ficheiros auxiliares (ficheiros CSV). Em termos funcionais, as expressões regulares não operam sobre valores externos, havendo a necessidade de capturar o valor da expressão e comparar o mesmo com o valor tido em ficheiro.

Inicialmente obtém-se o valor que se pretende analisar como apresentado na Figura D.5.

```
<ereg regexp="sip:([^@]+)@([^ ]+)"
  search_in="hdr"
   header="From:"
   check_it="true"
   assign_to="all,user,domain">
 </ereg>
```

Figura D.5: Obtenção do user-part e host-part do cabeçalho From

Após obtenção dos valores que se pretende validar, compara-se os mesmos com o valores do ficheiro (Figura D.6). Sobre o resultado da comparação realizada aplica-se a expressão regular apresentada na Figura D.7 de forma a verificar se o teste ocorreu com sucesso.

```
<assignstr assign_to="fileUser" value="[field0]" />
<assignstr assign_to="testUser" value="[$user]" />

<assignstr assign_to="fileDomain" value="[field1]" />
<assignstr assign_to="testDomain" value="[$domain]" />

<strcmp assign_to="result" variable="fileUser" variable2="testUser" />
<strcmp assign_to="result2" variable="fileDomain" variable2="testDomain" />
```

Figura D.6: Comparação dos valores anteriormente obtidos com valores de ficheiro

```
<ereg regexp="[-+]?([0-9]*\.[0-9]+|[0-9]+)" search_in="var" variable="result"
assign_to="result" check_it_inverse="true" />

 <ereg regexp="[-+]?([0-9]*\.[0-9]+|[0-9]+)" search_in="var" variable="result2"
assign_to="result2" check_it_inverse="true" />
```

Figura D.7: Validação dos valores obtidos da comparação dos valores dinâmicos

Bibliography

[1] Keith Knightson; Naotaka Morita; Thomas Towle. NGN Architecture: Generic Principles,Functional Architecture, and Implementation. *IEEE Communications Magazine*, 2005.

[2] Kai-Di Chang; Chi-Yuan Chen; Jiann-Liang Chen; Han-Chieh Chao. Challenges to Next Generation Services in IP Multimedia Subsystem. *Journal of Information Processing Systems*, 2010.

[3] Mobilecomms-technology. Softbank Mobile Corp, IMS Network, Japan. `http://www.mobilecomms-technology.com/projects/softbank/`, 2006.

[4] IETF. RFC 4566: SDP: Session Description Protocol, July 2006.

[5] ITU-T. Y.2001: General overview of NGN, December 2004.

[6] ITU-T. Y.2011: General principles and general reference model for Next Generation Networks, October 2004.

[7] ETSI. TR 180 001: NGN Release 1: Release definition, March 2006.

[8] 3GPP. TS 23.228: Service requirements for the IP Multimedia Core Network Subsystem, November 2000.

[9] IETF. RFC 3261: SIP: Session Initiation Protocol, June 2002.

[10] IETF. RFC 6733: Diameter Base Protocol, October 2012.

[11] IETF. RFC 3550: RTP: A Transport Protocol for Real-Time Applications, July 2003.

[12] David Gonçalves; Pedro Sousa; António Amaral; António Costa. Automatização de Testes SIP (Accept). *13ª Conferência sobre Redes de Computadores,*, October 2013.

[13] Mary Bellis. The History of the Electric Telegraph and Telegraphy. http://inventors.about.com/od/indrevolution/a/telegraph_2.htm.

[14] A.A. Gokhale. *Introduction To Telecommunications*. Thomson/Delmar Learning, 2004.

[15] Leonard C. Bruno. The Invention of the Telegraph. http://www.learnnc.org/lp/editions/nchist-antebellum/5423.

[16] Alexander Graham Bell's. Telegraphy Patent, March 1876.

[17] American Treasures of the Library of Congress. "Mr. Watson – come here!". http://www.loc.gov/exhibits/treasures/trr002.html.

[18] Adriano Moreira. Comunicações Sem fios - Evolução. Technical report, Universidade do Minho.

[19] Adriano Moreira. Redes Móveis Celulares. Technical report, Universidade do Minho.

[20] I. Brodsky. *The History of Wireless: How Creative Minds Produced Technology for the Masses*. Telescope Books, 2008.

[21] Sean Macaulay. The Cellphone Turns 40: Remembering Martin Cooper's Historic Call. http://www.thedailybeast.com/articles/2013/04/03/the-cell-phone-turns-40-remembering-martin-cooper-s-historic-call.html.

[22] Dr. Yunfei; Sengupta Dr. Jyotsna; Divya Kumar, Amit; Liu. Evolution of Mobile Wireless Communication Networks: 1G to 4G. *International Journal on Electronics & Communication Technology*, 2010.

[23] Communication Networks Research Lab. IS-41 Overview. http://cnr.kaist.ac.kr/lecture/te712_2001/download/05_IS-41.pdf.

[24] Ghassem Koleyni. Evolution to NGN. Technical report, ITU-T, 2005.

[25] ETSI. Digital cellular telecommunications system (Phase 2+);GSM Network functions, May 1996.

[26] Ching-Yung Lin; Jyh-Ren Shieh. IS-95 North American Standard-A CDMA Based Digital Celluar System.

[27] Adriano Moreira. Redes Móveis Celulares: GSM. Technical report, Universidade do Minho.

[28] Jochen H. Schiller. Mobile Communications Chapter 4: Wireless Telecommunication Systems. Technical report, UFreie Universität Berlin.

[29] Stanley Morgan. Economy + Internet Trends . `http://www.cs.rutgers.edu/~badri/552dir/papers/intro/MS_Economy_Internet_Trends_102009_FINAL.pdf`, October 2009.

[30] Ericson. Next Generation 112: Evolution of telecommunication. Technical report, 2010.

[31] ITU-T. ITU paves way for next-generation 4G mobile technologies. `http://www.itu.int/net/pressoffice/press_releases/2010/40.aspx#.UdRrRPl2WwU`, 2010.

[32] Chae-Sub Lee; Dick Knight. Realization of the Next-Generation Network. *IEEE Communications Magazine*, 2005.

[33] ITU-T. E.800: Quality of telecommunication services: concepts, models, objectives and dependability planning – Terms and definitions related to the quality of telecommunication services , September 2008.

[34] TISPAN. Defining the Next Generation Network. `http://www.etsi.org/tispan/`, 2008.

[35] ETSI. ES 282 001: TISPAN; NGN Functional Architecture Release 1 , August 2005.

[36] ETSI. TS 182 027: TISPAN; IPTV Architecture; IPTV functions supported by the IMS subsystem, 2008.

[37] 3GPP. TS 23.002 V5.12.0: 3GPP; Network architecture, September 2003.

[38] Antonio Cuevas; Jose Ignacio Moreno; Pablo Vidales; Hans Einsiedler. The IMS Service Platform: A Solution for Next-Generation Network Operators to Be More than Bit Pipes. *IEEE Communications Magazine*, 2006.

[39] G. Camarillo ; García-Martín. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*. Wiley, 2004.

[40] 3GPP2. About 3GPP2. `http://www.3gpp2.org/Public_html/Misc/AboutHome.cfm`, 2012.

[41] CableLabs. CableLabs Revolutionizing Cable Technology. `http://www.cablelabs.com/`, 2013.

[42] 3GPP. TS 23.002 V12.1.0: 3GPP; Network architecture, December 2012.

[43] Martin Koukal; Robert Bestak. Architecture of IP Multimedia Subsystem. *48th International Symposium ELMAR*, 2006.

[44] 3GPP. TS 23.167 version 11.7.0: 3GPP; UMTS; LTE; IP Multimedia Subsystem (IMS) emergency sessions, April 2013.

[45] 3GPP. TS 23.003 V11.5.0: 3GPP; Numbering, addressing and identification, April 2013.

[46] IETF. RFC 3398: Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping, December 2002.

[47] Filipe Leitão. Cenários de convivência de serviços de mensagens entre redes tradicionais e redes de próxima geração. Master's thesis, Universidade do Minho, Braga, 2009.

[48] PT Inovação. SIP Avançado: Estrutura SIP, June 2013.

[49] IETF. RFC 2543: SIP: Session Initiation Protocol , March 1999.

[50] IETF. RFC 3515: The Session Initiation Protocol (SIP) Refer Method , April 2003.

[51] IETF. RFC 3428: Session Initiation Protocol (SIP) Extension for Instant Messaging , December 2002.

[52] IETF. RFC 3263: Session Initiation Protocol (SIP): Locating SIP Servers , June 2002.

[53] IETF. RFC 5234: Augmented BNF for Syntax Specifications: ABNF , January 2008.

[54] IETF. RFC 2865: Remote Authentication Dial In User Service (RADIUS), June 2000.

[55] Anna Hosia. Comparison between RADIUS and Diameter. Technical report, Helsinki University of Technology, 2003.

[56] IETF. RFC 4006: Diameter Credit-Control Application, August 2005.

[57] IETF. RFC 4004: Diameter Mobile IPv4 Application, August 2005.

[58] IETF. RFC 4740: Diameter Session Initiation Protocol (SIP) Application, November 2006.

[59] Acme Packet. Acme Packet® Net-Net Session Director. http://www.acmepacket.com/collateral/acm/datasheet/APKT_DS_SD.pdf.

[60] BIG-IP. BIG-IP® Systems: Getting Started Guide. http://support.f5.com/content/kb/en-us/products/big-ip_ltm/manuals/product/bigip_getting_started_guide_10_1_0/_jcr_content/pdfAttach/download/file.res/BIG-IP_Systems__Getting_Started_Guide.pdf.

[61] BIG-IP. Configuration Guide for Local Traffic Management. http://support.f5.com/content/kb/en-us/products/big-ip_ltm/manuals/product/bigip9_0config/_jcr_content/pdfAttach/download/file.res/Configuration_Guide_for_BIG-IP_Local_Traffic_Management,_version_9.0.pdf.

[62] Fraunhofer FOKUS. Links for IMS Developers. http://www.openimscore.org/links.

[63] Richard Gayraud. Welcome to SIPp. http://sipp.sourceforge.net/.

[64] Sebastien Blavier; Simon Tatham. ExtraPuTTY. http://www.extraputty.com/.

[65] lua Documentarion. http://www.lua.org/docs.html.

[66] Catalina Oancea. pcap2sipp. http://pcap2sipp.sourceforge.net/.

[67] Digium. sniff2sipp. http://svn.digium.com/svn/sniff2sipp/trunk/.